An Argus Specialist Publication

FEBRUARY 1985

85p

NEW

Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

THE PLUS/4 SOFTWARE QUARTET—ARE THERE STRINGS ATTACHED?



MUSIC MAKING INSIDE

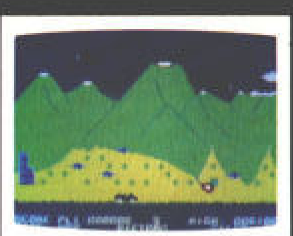
MICRO SOUND 64

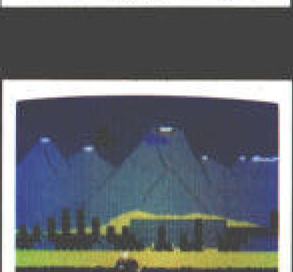
MUSIC MAKER

SPEECH SYNTHESISERS—ARE THEY ALL TALK?

A GREAT GAME FOR THE C16


Battle throughTime
IAN ELLERY
ANIROG
Battle throughTime
COMMODORE 64





Our COMMENT

Heed the editor's warning about computers that talk too much.

I have a dream. Well, it's more of a nightmare really. I am awoken in the morning by my alarm clock but instead of an insistant and irritating buzz it is programmed to rouse me from sleep in sweet and pleasant tones. I abuse it soundly. No longer do I drive to work in a grumpy and half-awake silence. Instead, I have to listen to the prattle of my car as it asks how I slept, what sort of evening I spent (I was talked at by the cathode ray tube) and what I ate for breakfast. My word-processor is so called because it talks all day. Its vocabulary is larger than mine (no cheap jokes please). My entire day is spent listening to an artificially intelligent babble.

It's started already, of course, and with the Commodore 64. We all know that the 64 has one of the best sound generating chips of any computer on the home market. Soon it may never tire of telling us so. However for the moment it is still finding its voice.

So, what are the tell-tale signs, the first faltering syllables? Not Mummy or Daddy, oh no, but "He slimed me" and "Destroy him my robots!". These immortal words appear on two of the best games on the market at the moment, Ghostbusters and Impossible Mission. Soon all computer games won't just be doing the nasty stuff to you but recounting all the gory details as well.

Fortunately, we do still have some control over the micro mouth. Speech synthesizers may be the computer's voice box but for the time being we tell them what to say. Our review of those speech synthesizers in this issue will show you how your 64 is already loosening up its larynx. Be warned.

Now, you're probably saying that I'm slightly paranoid. It's fun to hear your computer talk, to create different characters in your games and far more friendly to be told by your computer to do something rather than to read it on the screen. But already from America we hear stories of automatic doors expressing the wish that those who pass through them will 'have a nice day, now'. When I'm in the habit of talking to doors, I shall be grateful for the greeting.

Of course, my tongue is in my cheek. Better that than to be tongue tied by a talking computer. It might just not be the ghost in the machine which has the last laugh (see Ghostbusters) but the machine itself!

And what of my dream? Like all such visions it passes at the first sound of the alarm clock. I shout at the clock. After all, it is one of those new voice recognition ones.

Your comment

We're overwhelmed. By what? By you, of course. We called for reviewers and you answered, in your hundreds. We asked for your comments and you sent them to us. Not everyone could be a reviewer, unfortunately, there's just not enough software to go round. But we read all the reviews and enjoyed them all; the standard was so high that it made the task of choosing almost impossible. If you haven't heard yet, please bear with us as we're still trying to let everyone know whether they have been accepted.

Similarly, if you've written to us about the magazine, or with a problem or just about Commodore computing in general, we haven't forgotten you. It may be some time before we get round to replying but you can be sure that we have taken note of what you have to say. For example, Derren Nugent of Swindon writes: "By the way, could you print one or two C16 programs in your stunning mag...P.S. Your magazine is probably the best, most economical and informative mag for Commodore users...." I don't know about the 'probably', but how could we refuse a request like that. Derren, turn to page 34....



FEATURES



NUMBER 5
FEBRUARY 1985

Editor: Wendy J Palmer
Deputy Editor: Kevin Cox
Editorial Assistant: Alison Hjul
Advertisement Manager: Mike Segrue
Advertisement Copy Control: Laura Champion
Chairman: Jim Connell
Origination: Ebony Typesetiing
Design: MM Design

Editorial & Advertisement Office
No 1 Golden Square,
London W1R 3AB
Telephone: 01-437 0626
Telex: 8811896

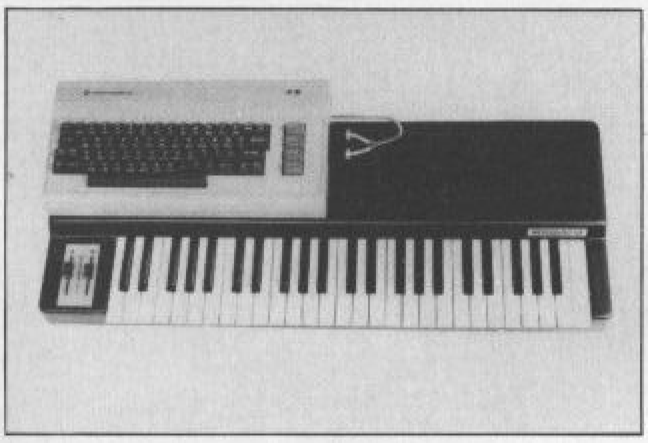
Your Commodore is a monthly magazine appearing on the first Friday of each month.

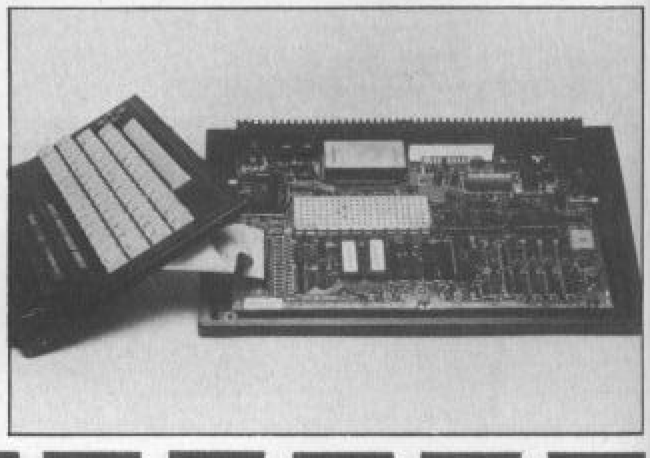
Distribution by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS. Printed by: Alabaster Passmore & Sons Ltd, Tovil, Maidstone, Kent.

Subscription rates upon application to Your Commodore Subscriptions Department, Infonet Ltd, Times House, 179 The Marlowes, Hemel Hempstead, Herts. HP1 1BB.









CONTENTS

SERIES

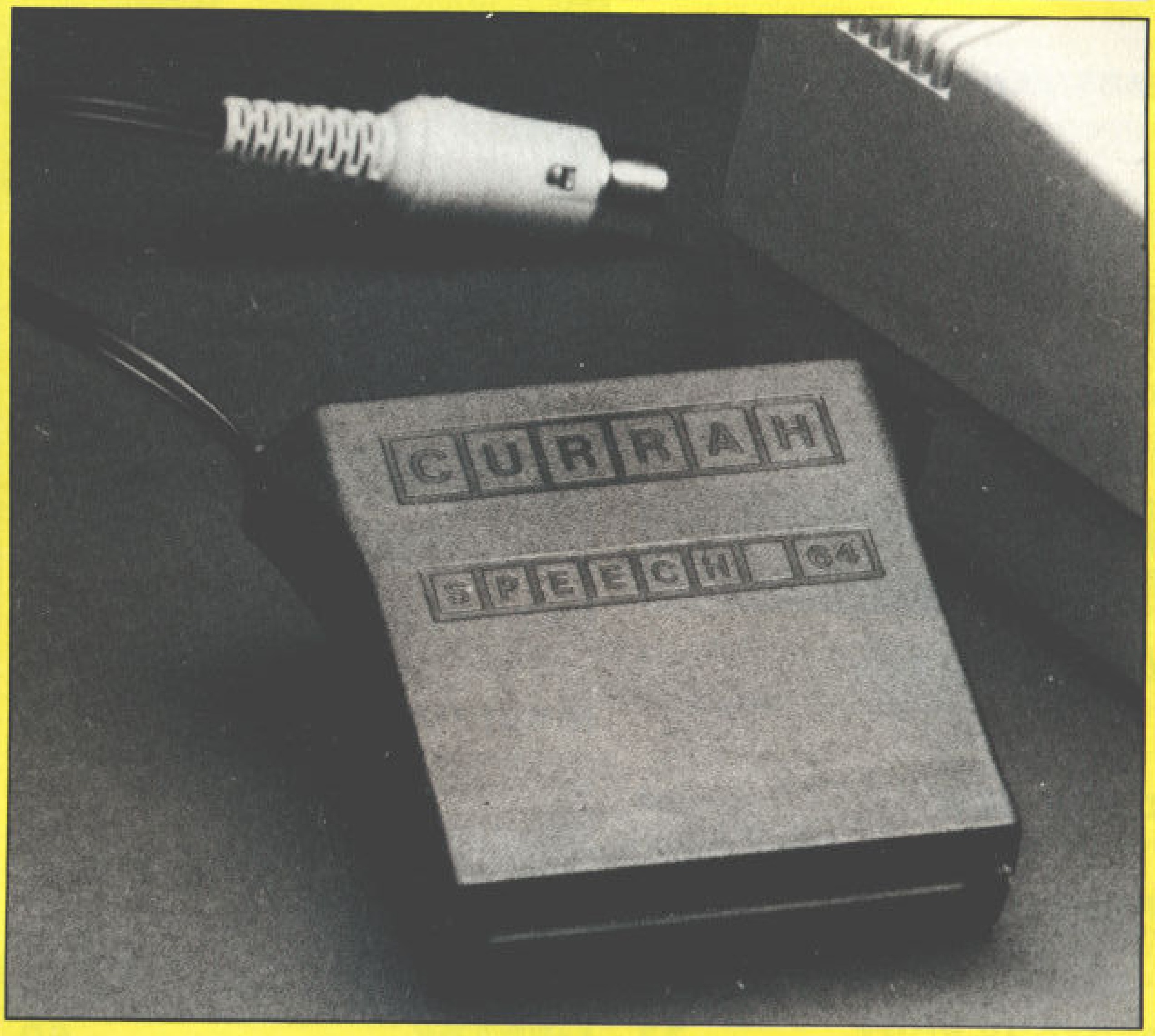
GAMES AND UTILITIES



REGULARS

Let your 64 answer back with some sound advice from David Crisp.

SPEAKER'S CORNER



THE CURRAH SPEECH UNIT IS a small black box which plugs into the cartridge port of your Commodore with a five pin DIN plug which then plugs into the audio/video socket. Two problems arise from the above arrangement. First if you use a Commodore monitor then your audio/video socket is in use and you would first need to wire the speech unit into the monitor plug. Not a fantastic alternative as this means that your speech unit is permanently attached to the monitor. It is not an insurmountable problem but will require a bit of work with cutters and a soldering iron. The second problem arises if you have an SX-64. This once again needs cutters and a soldering iron. The audio video socket and the cartridge port are a few inches further away on the SX and so it is necessary to splice in a couple of inches of extra wire between the plug and speech unit in order to plug it in. It is only a couple of inches but still a nuisance. If you have a standard 64 and use a TV as opposed to a monitor then it is a simple plug in job and this will apply to the majority so, in most cases, the above is not relevant.

Once it is plugged in it is very easy to use. I had my five year old daughter using it in just a couple of minutes. Before now the only speech units I had used were for the Spectrum and the BBC and to activate them it was necessary to look up the sounds you wanted to say, note the number of the sound, and then put them into some form of a list. What appeared on screen was simply a list of meaningless numbers which was hell's own job to edit into a reasonable sound. Not like that with the Currah. First you activate the unit by typing INIT, then to make it talk you simply type SAY" THE WORDS YOU REQUIRE" and to turn it off you type KOFF. It really is as easy as that. Those commands can be put into a program or used as direct commands.

Talk to Me

If you type in as a direct command INIT the speech unit will SAY the letters that you type from the keyboard. All were understandable except for the return key which to me sounded like a muppet with a speech impediment saying WRITTEN. And why, oh why, the dreaded American ZEEE when I am sure most people in this country say ZED. Those things apart it was smashing. It is a Robotic voice and not human sounding but is still very easy to understand. There are two voices one high and one low and intonation is possible but not quite so easy to get right. A very clever piece of programming has put Rules of speech into the unit and in the majority of cases it works exceptionally well. By rules of speech I mean that if you type SAY "MEEN" and then SAY "MEAN" they both sound the same (as they should). Very clever that, and effective. Currah say the way to get the most realistic sounds is to extend the basic SAY command to include mnemonics for particular sounds. It does work but I found it easier to write the word as it sounds e.g. To say Hawaii they suggest typing SAY "haw(ii)(ee)". This works very well but I found SAY 'HUYE" faster and easier to work out with almost as good a result. This only goes to show how versatile this unit is. There is a speech buffer which holds about thirty seconds worth of speech which means the program can carry on while speech is output. It is also possible to program the unit through machine code. Currah say it is difficult and so I tried. Believe me it is. It is so easy to use through BASIC it would be easier to exit from the coded program, return to BASIC, say what you have to and then return to code.

To sum up then, this unit is very easy to use and easy to get

good results from. How useful or educational to it is, is difficult to say but without doubt it is a lot of fun. I keep plugging it in and having another bash.

All the speech units I have heard have been male voices. I would like to hear one with a female voice.

Jamar Speakeasy

Another speech unit has just arrived and this one is made by a Yorkshire company. The Jamar Speakeasy is a self contained box which, in the case of the 64, is plugged into the user port. It is claimed to be compatible with a long list of machines (see list at end) with the relevant interface cable. It is a two tone grey box with a built in speaker and has a slightly cheap plasticky feel. On my review module the speaker grille was messy as many of the slots were gummed up with resin. The interface cable was a short piece of ribbon cable with an IDC type connector at one end. The other end was a plug to match the user port. This plug was meant for Chassis Mounting and so the wires of the ribbon were exposed. It is possible to get covered plugs and so I cannot see why the correct plug has not been used. This is a small point but reflects the general cheap finish of the unit.

Back to the book

Before I plugged the module in I did, as always, fully read the instruction manual. The very small print and the cheap paper put me off but I persevered. Horror of horrors this unit requires a mass of meaningless numbers in order to get anything out of it. On the Currah unit to say 'Computer' you type SAY"COMPUTER" however on the Jamar unit you type "8,24,16,9,49,31,13,52,4".

Which do you prefer. I soon made up my mind. I eventually forced myself to plug in this unit and blast away. It was first required that a short program was typed in to set the speech unit up. This done, I typed in a few of the numbers needed to make it say simple words. Chatter away it did and another few numbers made a few more words. After the ease with which it was possible to get the Currah going, this number method became a definite bore. Constantly referring to the manual for the appropriate number to say the sound is a long winded procedure. I know that not long ago the numeric way was the **only** way to get speech out, But the Currah shows that this is no longer the case. I tried to make up new words which were not in the manual. It can be done but believe me it does take a long time. Because there is no meaning to the numbers I feel that it would even be difficult to learn which number made the appropriate sound even with a lot of use.

Talking OK

All said and done, the quality of the speech was OK. Quite good in fact but still not good enough to make me go for this one as opposed to the Currah. The manual and the finish are not a patch on the Currah and at least the Currah did work first time. (The first Jamar did not work and so I had to wait for a replacement).

The Jamar unit costs about £5 more than the Currah unit and you still need to pay another £4.50 for the interface lead. So in the end you pay £10 more for a less desirable unit. The following appraisal is a comparison between the two units.

SPEECH 64
COMPATABILITY
No other machine though other versions of the Currah may be purchased.
VALUE FOR MONEY..★★★★★
EASE OF USE...★★★★
DOCUMENTATION...★★★
OVERALL...★★★★★
PRICE.........£29.95

SPEAKEASY
COMPATABILITY, BBC, DRAGON, ORIC, SHARP, VIC 20, MEMOTECH COLOUR GENIE (all with relevant interface cable)
VALUE FOR MONEY..★★
EASE OF USE...★
DOCUMENTATION...★★
OVERALL...★★
PRICE..£34.45 Inc. VAT + £4.50 for interface cable. TOTAL £38.95




A L I E N
In space no one can hear you scream.
NAVIGATOR
Shy, Skilful and Intelligent — Panics Easily.
EXECUTIVE OFFICER
Direct, Imaginative, Cautious, Loyal.
SCIENCE OFFICER
Secretive, Unlikeable, Brilliant — Occasionally Illogical.
CAPTAIN
Solid, Dependable, Courageous — Excellent Leader.
ENGINEERING OFFICER
Physically Strong, Low I.Q. Potentially Rebellious
3RD OFFICER
Wilful, Ambitious, Authoratitive, Resourceful.
ENGINEERING OFFICER
Cynical, Rebellious, Untrustworthy, Unflappable.
THE CREW
Personnel files follow — yours to command — well almost...
MIND GAMES
SPECTRUM 48K · CBM64
Argus Press Software Group
Featuring the unique Personality Control System
£8.99
No. 1 Golden Square, London W1R 3AB. Telephone 01-437 0626

Data Statements

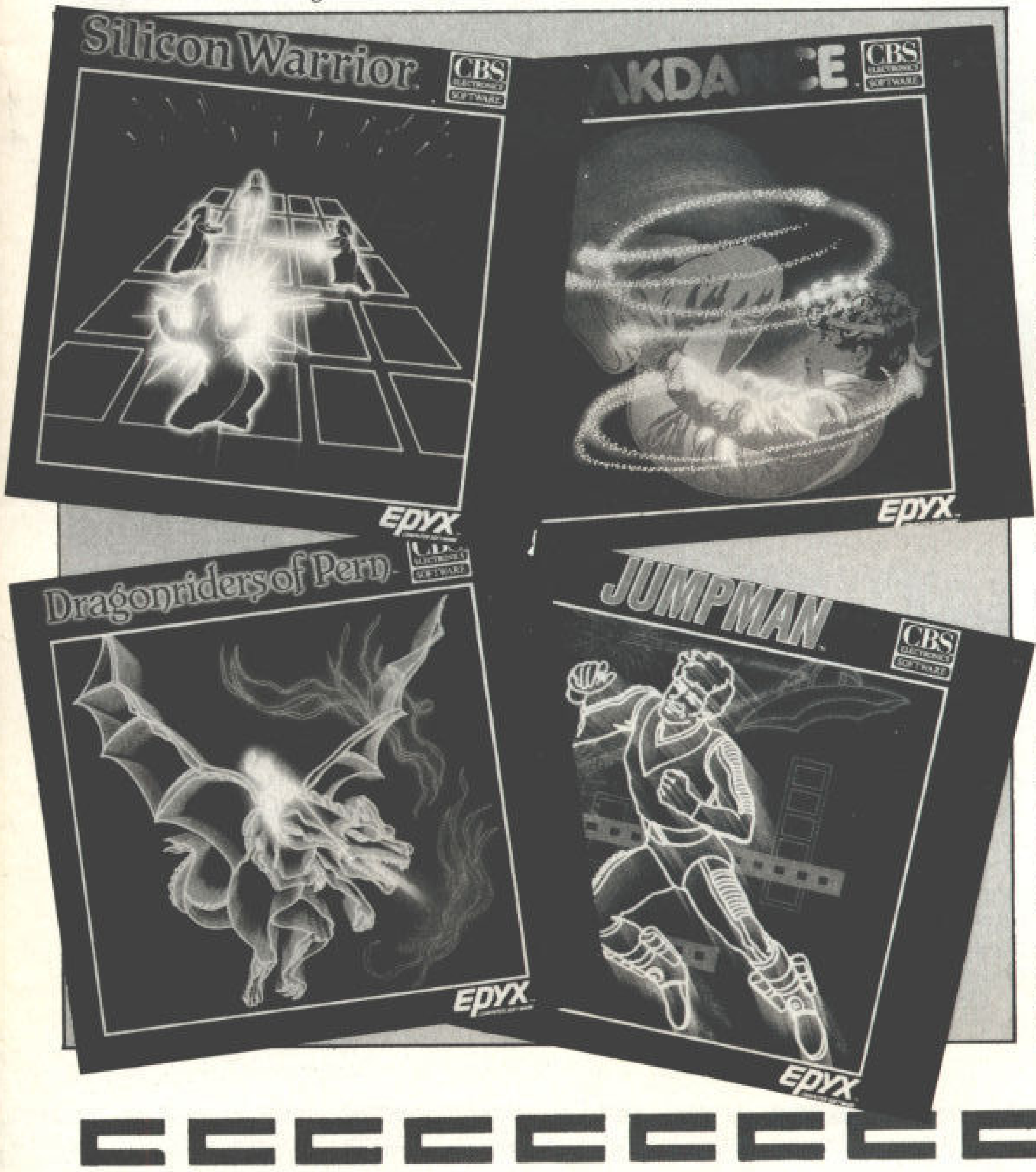
Off the record

TWO BIG RECORD COMPANIES ARE making a song and dance about entering the software market. They are CBS and Ariolasoft which is connected to Arista Records. The initial push for both companies is for the Commodore 64. For the time being, however, they are importing existing software from the US.

CBS have started rather modestly with 8 titles eg, Breakdance based on the latest dancing craze, and Impossible Mission, an arcade, adventure, strategy game of mind-boggling complexity — and it's got brilliant graphics. All are from Epyx (who are famous for Summer Games) and more of their titles are to follow.

There was nothing modest about Ariolasoft's launch. They hired the Hippodrome in London's West End for the night to tell the world about their first twelve titles. All imports again but they include at least two US Number Ones. Some are now getting a bit long in the tooth, like Lode Runner but the selection does include some brilliant games.

Fortunately, neither company rules out the possibility of commissioning home-grown software and with their funds and knowledge of a very similar market they look assured of success. Whether other software houses will be dancing to their tune in the near future remains to be seen.



CRL in fine voice

CRL'S LATEST PROGRAM IS SPRINKLED with liberal doses of culture. Entitled Tristan and Isolde, and bearing little resemblace to Wagner's opera of the same name, it should be available around the middle of December.

It was written by a 21 year old student at Heidelberg University; and CRL have been given the rights to market the product in English speaking countries.

The program will come with a booklet and will cost £8.95.

CRL are at CRL House, 9 Kings Yard, Carpenter's Road, London E15 2HD; tel. 01-533 2918.

Up a gumtree

UPPER GUMTREE IS A GRAPHIC ADVENTURE by Peter Cooke from Richard Shepherd Software which also includes an element of detective work.

Upper Gumtree is, apparently, the most boring place in the world until strange and terrible things begin to happen. When Professor Blowitovitz moves in with a lorryload of ominous looking gadgets, your aim is to uncover the professor's evil plot to end civilisation as we know it! Working in the dark, you must unravel the mystery, discover the threat to civilisation, find a solution and cope with the professor.

This game is littered with a selection of interesting characters apart from the dubious professor. Your attempts to save the world may be eased — by questioning them.

Time plays an integral role in the scoring system of Upper Gumtree. Richard Shepherd Software intend to present certificates (signed by Peter, Cooke and Richard Shepherd) to ten Upper Gumtree players who have completed the adventure in the shortest Gumtree Game Time.

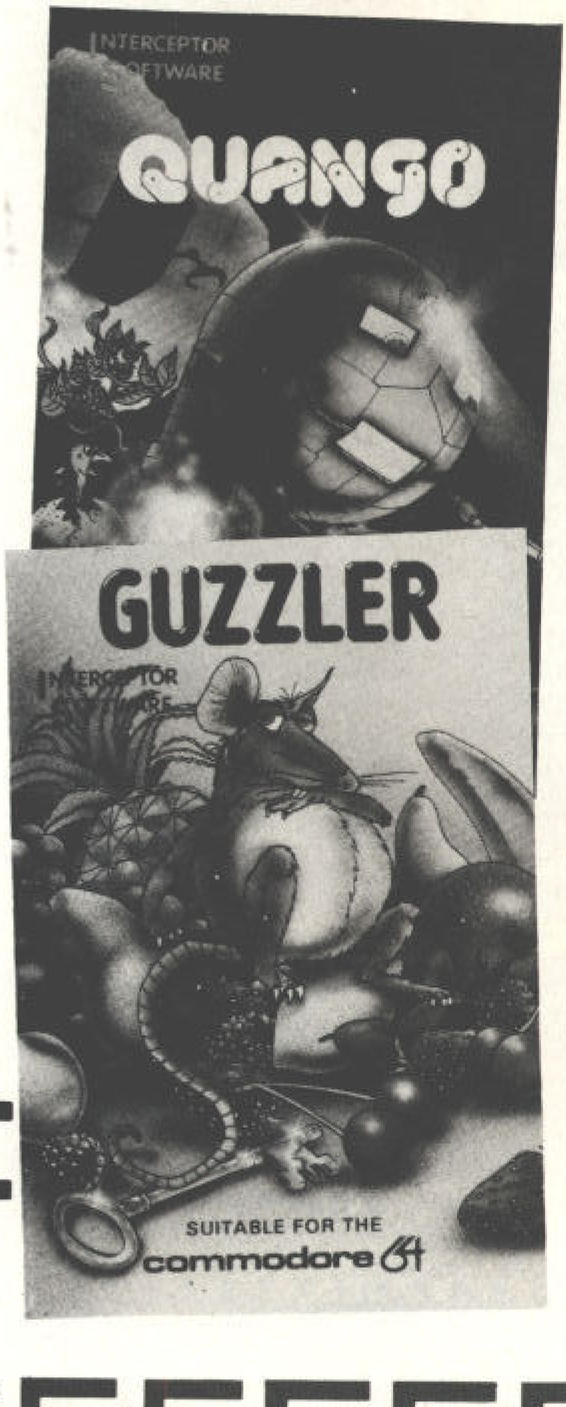
Upper Gumtree will be available on tape initially at £6.50. Richard Shepherd Software can be contacted at Elm House, 23-25 Elmshott Lane, Slough, Berkshire; tel. 06286-63531.

Data Statements

High speed games

INTERCEPTOR SOFTWARE ARE TO RElease two new titles for the Commodore 64 — announced their new Microload system — a high speed loading technique for the Commodore 64. Interceptor claim that this new system loads programs into the Commodore 64 at around 3000 baud as well as displaying a smooth scrolling screen message, a high resolution colour picture and plays music — all while the program is loading.

Interceptor Software can be contacted at Lindon House, The Green, Tadley, Hants; tel. 07356-71145/3711.

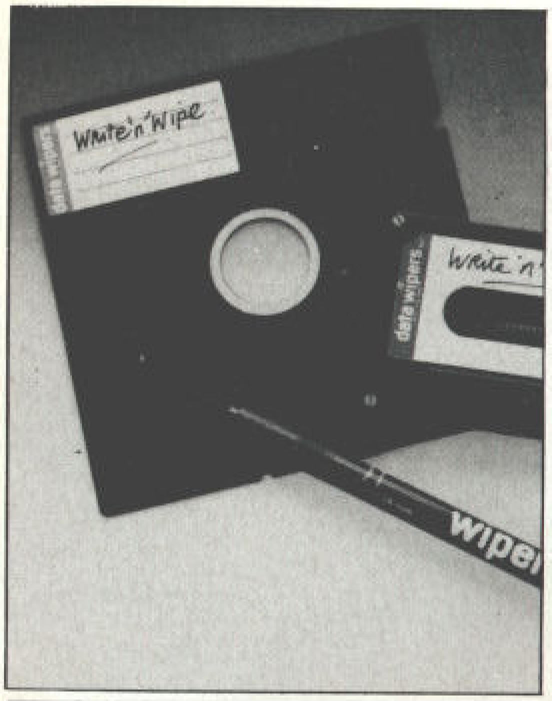


Tidier titles

WARNES WIPERS WIPE CLEAN COMPuter labelling kits claim to enable the user to title their programs (on cassette or disc) simply and clearly thus eliminating the mess you get with the pencil/eraser combination. To alter the title, you wipe the label clean with a damp tissue and retitle.

The kits include 12 specially designed labels, the WIPERS pen and a set of alphanumeric mini labels for indexing and permanent information.

WIPERS Computer labels cost £1.49. Warnes WIPERS are at 23 Werter Road, London SW15 2LL; tel. 01-788 1782.



Heroic adventure

SPIDERMAN, THE LATEST SCOTT Adams' adventure from Adventure International, was delivered safely into the hands of Micro Dealer (UK) Ltd. (who will be distributing it) by Spiderman and The Incredible Hulk, at the end of November. Spiderman will be available on a number of home computers, including the Commodore 64. It will cost £9.95.



New software houses

TWO NEW SOFTWARE HOUSES HAVE been launched. They are Lionheart Software and Orpheus.

Lionheart Software was launched by a team headed by Mark Meakings, cofounder of Bubble Bus Software. The company hope to produce and publish a range of products for most makes of micro computer. Lionheart will be encouraging links with freelance programmers alongside producing inhouse products.

Orpheus has been initiated by three Oric Software personalities — Paul Kaufman (Managing Director), Cathie Burrell (Financial and Administrative Director), Geoff Phillips (Software Director). The company is based just outside Cambridge. It will initially launch three Oric 1/Atmos titles but these will be followed by titles for other machines, including the Commodore 64.

Lionheart software are based at 29, The Boundary, Langton Green, Tunbridge Wells, Kent, TN3 0YA. Orpheus are based at The Smithy, Unit 1, Church Farm, Hatley, St. George, Nr. Sandy, Beds, SG19 3HP.

Data Statements

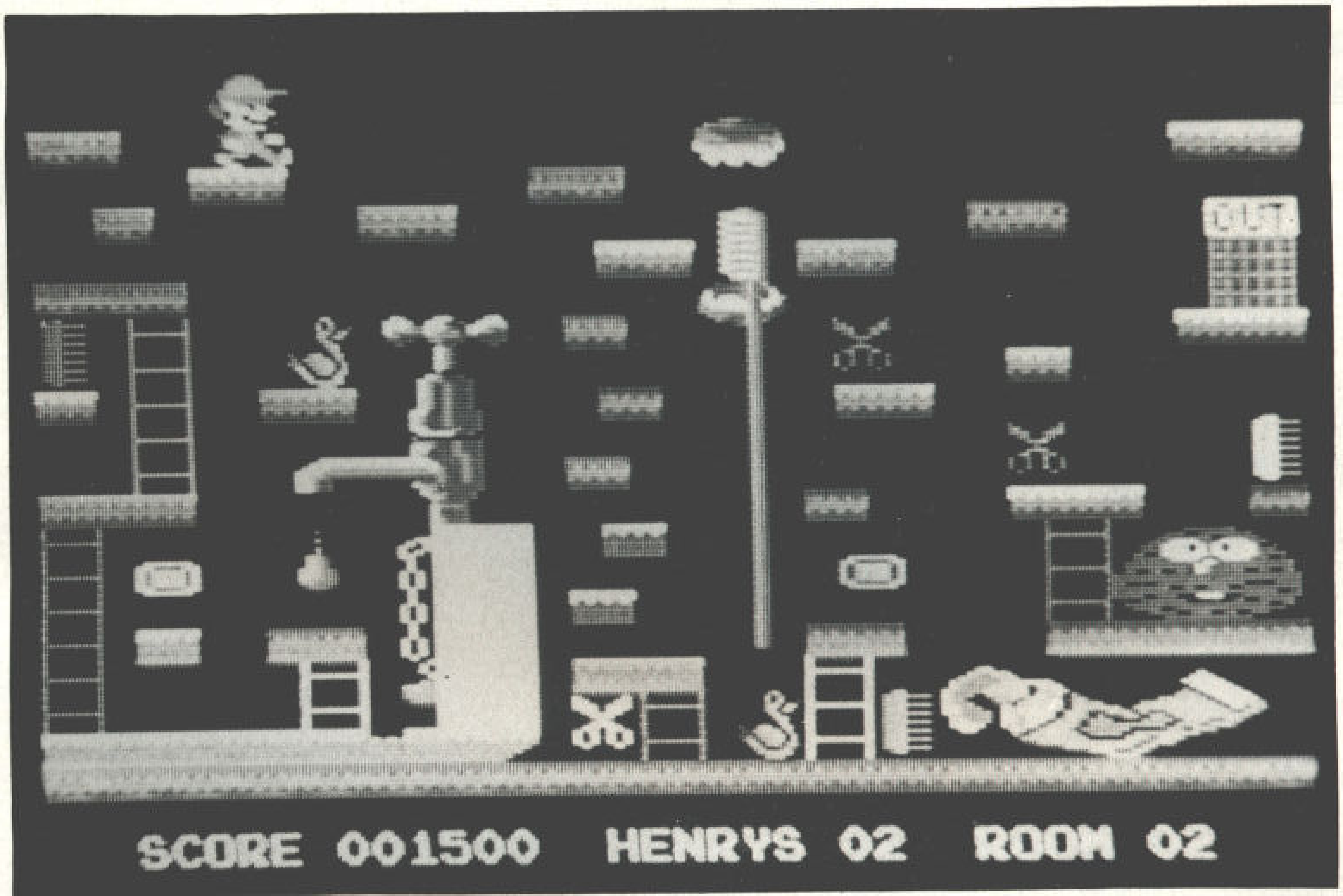
Best of British

PSYCHIC ENGINEERING CONTROL Group have designed Soldier of Fortune for the English Software Company. This is a multi-screen arcade adventure set in Spain in which you encounter windmills, treasures, snakes, bats, blood thirsty cave birds and lots lots more in your quest for gold.

Another new game from The English Software Company is Henry's House in which little Henry takes the player for a tour through his new home. The game features hi-res graphics and is written in 100% machine code.

Henry's House costs £8.95 on cassette. Both games are for the Commodore 64.

The English Software Company are at Box 43, Manchester, M60 3AD; tel. 061-835 1358.



Fleeing from the Fuzz

LONGMAN/SCHOLASTIC SOFTWARE have released Agent U.S.A, an arcade/strategy game for the Commodore 64. The player's aim as Agent U.S.A. is to save the U.S.A. from the rapidly advancing Fuzz after a Fuzzbomb lurking in a US city is infecting everyone who touches it and turning them into Fuzzbodies. You must race against time, building up your defences whilst checking on the advance of the Fuzz. Then you catch a train, speed into the heart of Fuzz territory, find the bomb and destroy it.

Agent U.S.A. has been produced by Tom Snyder, a top American games designer. It is available on cassette or disc for the Commodore 64 and retails at £7.95 on cassette and £14.95 on disc.

Longman Software can be contacted at Longman Group Ltd., Longman House, Burnt Mill, Harlow, Essex CM20 2JE; tel. 0279-26721.

New Mastertronic label

MASTERTRONIC HAS LAUNCHED A new adventure games label — Master Adventurer. Titles on this label will cost £2.99 and £3.99.

The first title to be released on the Commodore 64 is Se-Kaa of Assiah. Another game, Black Crystal, is due out for the Commodore 64 in the new year at a price of £3.99 (the package includes two cassettes and a booklet).

Mastertronic are at Park Lorne, Park Road, London NW8 7JL; tel. 01-492 3316.

Holiday competition

COMMODORE COMPUTERS AND THE Spectrum Group PLC are sponsoring a competition to win a holiday in Florida. The holiday package includes a visit to Disney World, a guided tour of the Kennedy Space Centre, a trip to Florida Keys and accomodation in the Sheraton World Hotel and Sheraton Sand Key Hotel as well as £2500 spending money. The winner can also take three friends. The holiday must be taken during the first ten months of 1985. The ten runners-up will receive £100 gift vouchers to spend in any Spectrum Group appointed dealer store in the United Kingdom.

The competition involves listing in order of importance seven features on the new Commodore Plus/4 and completing a tie-breaker.

Spectrum Group plc are at Hunting Gate, Hitchin, Herts, SG4 0TJ; tel. 0462-37171. Further information can also be gained from: Crosweller Publicity on 01-481-3333.

Slip Us a disc

LEVEL 9, THE ADVENTURE GAME SPECialists have just announced that all their games will be available soon in disc versions. Anyone who already owns the cassettes need not throw them away if they would rather have discs. Level 9 will send you the disc if you send them the cassette and the difference, which should be about £2.50.

Another innovation is their new catalogue. They have designed it as a full-colour A3 poster with the games described on the back. The picture on the front will be changed every couple of months, so if you want to start collecting, send a large stamped and self-addressed envelope to Level 9 now.

The address is Level 9 Computing, 229 Hughendedn Road, High Wycombe, Bucks HP13 5PG.





Educational fairy-tale

DATABASE PUBLICATIONS OF STOCKport have just released an educational adventure, 'The Magic Sword' for the Commodore 64. It contains a 48 page colour book to set the scene, as well as a software cassette which enables the child to participate in the subsequent adventure.

The child is transported into a magic world of a handsome prince, a beautiful princess, a castle with secret passages and dungeons, dense forest, deep caves and a crooked house in which there lives a wicked witch. The child has to travel through the countryside and explore the castle to find the princess and rescue her from the wicked witch. Various magic objects must be found to reach the end of the story.

All the text is in double height characters to make it easy to read and a compass on the screen throughout the game points out the directions in which moves are allowed.

The Magic Sword is available on the Commodore 64 for £8.95. Database Publications are at Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY; tel. 061-456 8383.

Data Statements

Broad minded

ARGUS PRESS SOFTWARE HAVE announced that their next game under the Mind Games title will be entitled Broad Street. It will be based on the new Paul McCartney film — Give My Regards to Broad Street. Its programmers are Concept Software — the creators of APS's last game, Alien.

Broad Street should be available on the Commodore 64 at the end of January.



PSS magic

PSS'S ADVENTURE GAME, SWORDS AND Sorcery, should be available on the Commodore 64 in January, 1985. The program will contain 86 monsters, 635 locations, an 800+ dictionary, almost 2,000 objects and over 4,000,000 ways to be insulted.

Mike Simpson, who is Swords and Sorcery's programmer, spent over 18 months planning and developing the program. He claims that the program will run in Real Time, will be fully menu driven and that the player will be able to control weapons during combat sequences. There are a series of quests within the game.

PSS are at 452 Stoney Stanton Road, Coventry, CV6 5DG; tel: 0203-667556.



First Publishing

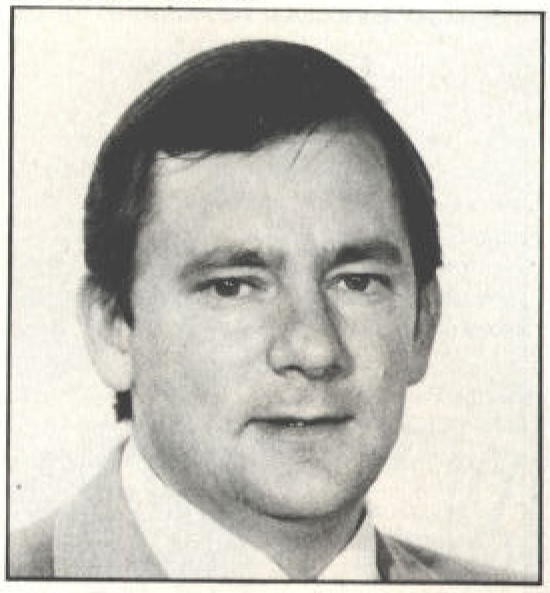
A NEW BRITISH PUBLISHING COMPANY, First Publishing Ltd., are launching a series of new books and software packages for the Commodore 64. They hope to have a total of 30 book titles and 15 software products on the market by Spring 1985. Their initial titles include 'Anatomy of the 64' and 'Tricks and Tips for the 64'. Their software products include 'FirstBASE 64' (a data manager) and 'FirstWORD 64' (a word processing package).

Data Statements

Taskset go to war

TASKSET LTD. HAVE INCORPORATED 8,192 screen locations into their latest game — Cadcam Warrior. The aim of the game is to find your way to Cadcam Control, the centre of this machine generated environment. The player must destroy his opponents and clear all the obstacles the computer has put in his way as quickly as possible. If the player does well, he may leap ahead to a further location rather than the next logical location, thus making each game slightly different from the previous one.

The game will be accompanied by a booklet including maps and information. It is expected to retail at £9.95 on cassette and £12.95 on disc.



New Commodore appointments

DAVID GERRARD, FORMER CONSUMER Marketing Consultant with Plessey, has been appointed as Commodore UK's Marketing Manager. His responsibilities include enhancing Commodore's dominant position in the home computer market, establishing Commodore's new products in this market and spearheading Commodore UK's assault on the business computer market in 1985.

Mike Beale, also from Plessey, has been appointed as Plant Manager at Commodore's new 200,000 square feet manufacturing facility in Corby, Northants. His job entails managing all manufacturing operations in Corby.

Frankie goes to sea

OCEAN SOFTWARE HAVE JOINED forces with Island Records and creative producers Zang Tumb Tumm to produce a Frankie Goes to Hollywood game. The band will receive a royalty from sales of the game. The companies involved see this as an opportunity to bring together previously isolated industries thus attracting a potentially wider purchasing audience.

Although details of the game have not yet been released, the new Frankie game should be available in the early Spring of 1985 with a retail price of £9.95.



Neat listings are easier to de-bug. Les Allan tidies up.

PROGRAM LISTER



MULTI-STATEMENTS IN BASIC programs separated by a colon can prove tedious to read especially when trying to de-bug a lengthy program. This utility, when resident high in memory, provides an alternative to the normal LIST. It uses the ★ [RETURN] key. The program prints each line number in reverse field (dark key) and thereafter each statement on a separate line (light grey). Scrolling is achieved with the cursor up or cursor down key and provision is made for single step or continuous update if the cursor key is held down. Pressing return cancels the current list.

The program as listed must be entered exactly as written and saved prior to running in case a typing error causes a system crash. Error trap routines have been incorporated so as to minimise that possibility. When completed insert a blank tape into the cassette, run the program and when prompted "HAVE YOU SAVED THIS PROGRAM Y/N" press Y. The machine code files Program Lister and Lister will now be saved to tape. Provision has been made in the program so that this utility can be saved and re-loaded from either tape or disk without any changes whatsoever.

Upon subsequent use simply type LOAD for tape or LOAD"PROGRAM LISTER",8,1 for disc and the saved program will automatically load and prepare the operating system to recognise the new command. Once loaded just type in SYS 49152. Should a Cold Start (SYS 64738) be made during normal operation the new command must be re-activated by SYS 49152.

A comparison of the finished listing is shown below:

Program Listing

```
10 REM **** PROGRAM LISTER FOR COMMODORE 64 *****
15 :
20 POKE53280,12:POKE53281,12
25 PRINTCHR$(147)CHR$(31)SPC(9)"%%% PROGRAM LISTER %%%"
30 PRINT:PRINT
35 PRINT"THIS MACHINE CODE PROGRAM ALLOWS LISTING"
40 PRINT"AND SCROLLING OF  BASIC  STATEMENTS WHEN"
45 PRINT"USED WITH THE * KEY. ALSO EACH STATEMENT"
50 PRINT"IS  PRINTED ON  A SEPARATE LINE. INDENTS"
55 PRINT"ARE PROVIDED DURING  FOR-NEXT  LOOPS AND"
60 PRINT"AFTER   IF-THEN  STATEMENTS  MAKING BASIC"
65 PRINT"LISTINGS OF PROGRAMS MUCH EASIER TO READ"
70 PRINT
75 PRINT"THIS PROGRAM INCLUDES A  M/C  ROUTINE TO"
80 PRINT"SAVE BOTH THE AUTO BOOT AND MAIN ROUTINE"
85 :
90 REM *************** HEX  LOADER **************
95 :
100 INC=0:SUM=0
105 READDA$:IFDA$="END"THEN165
110 IFLEN(DA$)<>2ANDDA$<>"END"THEN205
115 H=ASC(LEFT$(DA$,1)):H1=(H-48)*16:IFH>57THENH1=(H-55)*16
120 H=ASC(RIGHT$(DA$,1)):H2=(H-48):IFH>57THENH2=(H-55)
125 BCD=H1+H2:IFBCD<0ORBCD>255THEN205
130 POKE49152+INC,BCD:INC=INC+1:SUM=SUM+BCD
135 PRINT:PRINTSPC(5)"DATUM LEFT FOR TRANSFER :";
140 PRINT688-INCCHR$(157)CHR$(32)CHR$(145)CHR$(145)
145 GOTO105
150 :
155 REM ************** SAVE  ROUTINE *************
160 :
165 IFINC<>688ORSUM<>87290THEN205
170 PRINTCHR$(147)"DATA TRANSFER COMPLETE"
175 PRINT:PRINT:PRINT"HAVE YOU SAVED THIS PROGRAM Y/N"
180 GETKEY$:IFKEY$<>"Y"ANDKEY$<>"N"THEN180
185 IFKEY$="Y"THENSYS49744
```



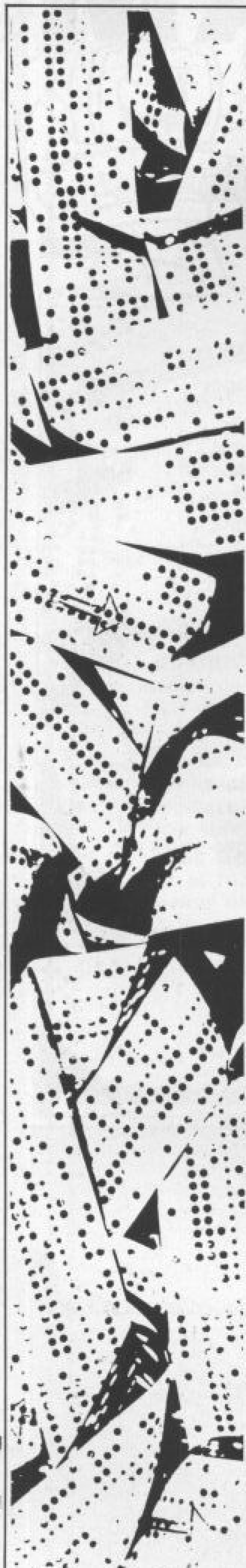
Enhanced List (★)

```
10
   FORA=1TO2:
      PRINT"PROGRAM LISTER"
20
      FORB=1TO2:
         PRINT"FOR COMMODORE 64"
30
         FORC=1TO2:
            PRINT"BY LES ALLAN"
40
            NEXT:
         NEXT:
      NEXT
50
   END
```

Normal List (LIST)

```
10 FORA=1TO2:PRINT"PROGRAM LISTER"
20 FORB=1TO2:PRINT"FOR COMMODORE 64"
30 FORC=1TO2:PRINT"BY LES ALLAN"
40 NEXT:NEXT:NEXT
50 END
```

Program Listing (cont.)



```
190 PRINTCHR$(17)CHR$(17)"SAVE"CHR$(34)"LISTER BOOT"CHR$(34)
195 PRINTCHR$(145)CHR$(145)CHR$(145)CHR$(145)
200 END
205 PRINTCHR$(147)"ERROR IN DATA STATEMENTS !!!":STOP
210 :
215 REM ******* HEX DATA FOR PROGRAM LISTER ******
220 :
225 DATA 4C,80,C1,00,A0,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA
230 DATA EA,EA,EA,A9,4C,85,7C,AD,22,C0,85,7D,AD,23,C0,85
235 DATA 7E,60,24,C0,C9,2A,D0,09,48,AD,7A,00,C9,00,F0,09
240 DATA 68,C9,3A,90,01,60,4C,80,00,A9,02,8D,FB,00,20,73
245 DATA 00,F0,0E,B0,15,20,6B,A9,20,60,C0,20,D7,AA,4C,2A
250 DATA A5,A9,00,85,14,85,15,18,90,EE,4C,08,AF,EA,EA,EA
255 DATA 20,13,A6,A0,02,B1,5F,85,14,C8,B1,5F,85,15,A0,00
260 DATA B1,5F,C9,00,D0,2F,C8,B1,5F,C9,00,D0,28,F0,45,A9
265 DATA 00,C5,14,D0,06,C5,15,F0,3B,C6,15,C6,14,20,13,A6
270 DATA A0,02,B1,5F,C5,14,D0,E7,C8,B1,5F,C5,15,D0,E0,20
275 DATA 44,E5,18,90,C9,20,EC,C0,20,E4,FF,C9,00,F0,F9,C9
280 DATA 91,F0,CC,C9,50,EA,EA,EA,C9,0D,F0,08,E6,14,D0,A0
285 DATA E6,15,D0,9C,60,38,E9,7F,AA,24,49,A0,FF,CA,F0,08
290 DATA C8,B9,9E,A0,10,FA,30,F5,C8,B9,9E,A0,30,06,20,E0
295 DATA C1,D0,F5,60,A4,49,29,7F,20,E0,C1,60,A0,02,20,D7
300 DATA AA,E6,C7,B1,5F,AA,C8,B1,5F,20,D0,C1,C6,C7,20,D7
305 DATA AA,A6,FB,20,73,C1,A9,00,85,FD,A0,03,C8,B1,5F,C9
310 DATA 00,F0,53,A6,FD,D0,04,C9,80,B0,1B,20,E0,C1,C9,22
315 DATA D0,08,48,A5,FD,49,01,85,FD,68,C9,3A,F0,26,D0,DC
320 DATA EA,EA,EA,EA,EA,EA,C9,82,D0,06,CE,FB,00,CE,FB,00
325 DATA 48,20,C5,C0,68,C9,81,F0,24,C9,A7,D0,BF,E6,FC,E6
330 DATA FC,18,90,B8,20,D7,AA,A9,00,85,FD,A5,FB,65,FC,AA
335 DATA 20,73,C1,18,90,A6,A9,00,85,FC,85,FD,60,E6,FB,E6
340 DATA FB,D0,99,E0,00,F0,07,20,3F,AB,CA,18,90,F5,60,EA
345 DATA A2,00,BD,A0,C1,F0,07,20,D2,FF,E8,4C,82,C1,AD,03
350 DATA C0,85,37,85,33,AD,04,C0,85,38,85,34,4C,10,C0,00
355 DATA 93,05,12,20,50,52,4F,47,52,41,4D,20,4C,49,53,54
360 DATA 45,52,20,42,59,20,4C,45,53,20,41,4C,4C,41,4E,20
365 DATA 92,9A,0D,0D,0D,0D,00,00,00,00,00,00,00,00,00,00
370 DATA CE,86,02,CE,86,02,20,CD,BD,EE,86,02,EE,86,02,60
375 DATA EE,86,02,20,D2,FF,CE,86,02,60,EE,EE,EE,EE,EE,EE
380 DATA A9,00,8D,20,D0,8D,21,D0,A9,01,A6,BA,A0,FF,20,BA
385 DATA FF,A9,06,A2,F0,A0,02,20,BD,FF,A9,00,A2,FF,A0,FF
390 DATA 20,D5,FF,A9,83,8D,02,03,A9,A4,8D,03,03,A9,0F,8D
395 DATA 20,D0,A9,06,8D,21,D0,20,00,C0,4C,94,E3,00,00,00
400 DATA 00,00,00,00,00,00,00,00,00,4C,49,53,54,45,52,00
405 DATA 00,00,00,00,00,00,00,00,00,8B,E3,A7,02,EE,EE,EE
410 DATA A2,00,BD,F0,C1,9D,A7,02,E8,E0,5D,D0,F5,A9,01,A6
415 DATA BA,A0,01,20,BA,FF,A9,0E,A2,A4,A0,C1,20,BD,FF,A9
420 DATA A7,85,2B,A9,02,85,2C,A2,04,A0,03,A9,2B,20,D8,FF
425 DATA A9,01,A6,BA,A0,01,20,BA,FF,A9,06,A2,AC,A0,C1,20
430 DATA BD,FF,A9,00,85,2B,A9,C0,85,2C,A2,F0,A0,C1,A9,2B
435 DATA 20,D8,FF,A9,01,85,2B,A9,08,85,2C,4C,CA,02,EE,EE
440 DATA END
445 :
450 :
455 ********************************************
460 *                                          *
465 *       PROGRAM LISTER FOR COMMODORE 64    *
470 *                                          *
475 *       UTILITY COMMANDS ARE AS FOLLOWS    *
480 *                                          *
485 *       SYS 49152  BOOTS LISTER ROUTINE    *
490 *                                          *
495 *       *   [RETURN]   ACTIVATE ROUTINE    *
500 *                                          *
505 *       *10 [RETURN]  LIST FROM LINE 10    *
510 *                                          *
515 *                                          *
520 *       LES ALLAN         19TH NOV 1984    *
525 *                                          *
530 ********************************************
READY.
```



If you've got any useful advice to give Your fellow Commodore readers or any problems with which you need our experts' help, put pen to paper.

INPUT

INPUT

I have successfully keyed in and saved the 'Turbo 64' program from the December 1984 edition of 'Your Commodore'. But, I find the directions in your magazine rather difficult to follow. Please could you explain how to load and save my ordinary BASIC programs to run in the 'Turbo 64' format.
J. R. O'Donnell
Kent

OUTPUT

We have received a lot of queries regarding this program so here is how to use it:
Type in the program exactly from the listing on pages 38 and 39 of the December issue, with the errata to lines 255 and 585 from last month's issue.
BEFORE RUNning the program, save it off under the name "turbo" or similar. Now RUN the program and it will read in the data. This will take a little time. Once this is done, you will be asked if this program has been saved. This question refers to the BASIC boot program which we saved off before RUNning this part. Therefore, our answer is "y" and press RETURN. The program now exists as one line when listed. Having done this, a prompt will appear on the screen. Press RECORD and PLAY and the finished program will be SAVEd as "T 64". Make sure you do not overwrite the BASIC loader program on the cassette when SAVEing as if something does not work you will not want to re-type the listing.
To use the utility type the following:

```
LOAD"T 64",1,1
RUN
SYS 64738
SYS 50000
```

The turbo mode is now in operation and all of the commands on page 38 should now work. Please note that any software must be SAVEd using this utility before it can be loaded using it. Thus you cannot use this utility to load your existing software until you have saved that software using the —S"file name" command.

INPUT

We purchased a Commodore 64 computer, with a display monitor, a VIC 1541 for a single drive floppy disc and an MPS-801 Dot Matrix Printer in order to computerise our accounts.

So far we have a program which enables us to print the accounts the patients and the statements to the nurses. This program was prepared by a friend.

Now we want to buy a program which will enable us to computerise all our accounts and records.

Will you please advise us as to the most appropriate program for this purpose.
Chalfont Nurses Ltd. Hertfordshire

OUTPUT

There are a number of accounts packages available for the 64. Take a trip to your local business computer shop and ask to see some of them demonstrated. The suit of programs produced by Anagram will certainly be a reasonable buy. Their programs started life on the Commodore business machines and the programs tend to be fast, flexible and easy to use. You can get in contact with them at:
Anagram Systems
60a Queen Street
Horsham West Sussex
RH13 5AD Telephone (0403) 59551

Note that if your accounts are large and complicated and you intend using your system for other applications, I would recommend buying a business computer now as Commodore business machines have the advantage of being well established and may be networked easily.

INPUT

I am trying to auto run my programs on a VIC 20 (I have a VIC 20 + 16K) from the LOAD (Return) command, that is auto run the first program on the cassette tape from within the program or tape header.

I have looked in books and magazines but have not found anything suitable for my use.

If you can suggest any routines or hints that I could include in my programs I would be very grateful for your help.
A. Orritt,
Merseyside

OUTPUT

This is possible by making the first program on the cassette load over the I/O vectors in page three. This program will have to be SAVEd as a type three tape file so that it is not relocated when LOADed. Make the close channel vector point at a piece of machine code loaded as part of the first program. This machine code can then jump to the BASIC load routine. This is quite tricky but will provide a reasonable level of program protection.

INPUT

I bought a Commodore 64 nearly a year ago to replace a Commodore PET. I have had an intermittent, but aggravating, problem with the Commodore 64 that has persisted despite replacing the computer. From time-to-time the Commodore initial screen is displayed with the complete loss of the content of memory. The problem is no doubt due, in part or entirely, to the power supply but fitting a surge suppressor has had little or no effect. I still have the PET and that runs and runs without any problem. The resetting of the machine occurs most frequently when in hi-res mode. Can you advise me of how I may cure this fault?
D. J. Clarke
Bristol

OUTPUT

A common and aggravating problem — my fridge causes the same. Expensive main suppressing (£200 worth!) made no difference. I suggest you move house or change all other electrical equipment or buy a new computer. Seriously though, it is sensitive to equipment with relays that switch in and out. Also it appears to be a problem only associated with the early 64s...Sorry.

OUTPUT

VIC GAMES PROGRAMMING

This month, Bryn Phillips shows you how to add a professional touch to your programs through, for example, memory expansion or stylish introductions.

HOWEVER GOOD YOUR idea for a program, and however well you write the main program, the whole thing can be easily spoilt by poor presentation. There's nothing worse when starting off a game, than to be met with an off-centred title, no instructions, and then to frantically try to steer your space ship out of the way of the space debris by stabbing blindly at keys, only to find the game has been written for a joystick! It's important to put these details into your game, together with refinements such as a high score facility, key/joystick option, replay facility, and if you can find room, a hall of fame.

In previous articles I've stressed the importance of tight programming — the more you squeeze it down the more you get in, with the added bonus of faster action. Unfortunately, however hard you try to squeeze down your program, you're still going to be hard pushed to put in all the features you might like. Just the title and instructions could take up 1-2K, depending on your ambitions. And then there's the clever graphics. The data statements alone start to eat into your valuable memory, and let's face it, on the unexpanded VIC 20 you've only got 3.5K. Or have you?

The two part program

The way round the problem is to write your program as a two part program.

Part 1
Initial title display
Instructions/scenario
Joystick/keyboard option
Customised graphics
etc

Part II
Game
High score display
Hall of fame
Score update
etc

Things are looking up. If you do this you will have converted your VIC 20 into an impressive 7K without adding any expansion. The only problem of course is that the first program is lost when you LOAD the second. At first sight this would make the whole thing pointless, apart from using the program to give the instructions for the game. When you lose the program, you also lose any values for the variables you have defined in the program. However there is a way round this problem. If you POKE the values for these variables in protected memory, using the technique outlined last month, they stay there, waiting to be pulled out in the next program. This is shown again in Listing 1. Line 10 moves BASIC to protect an area of memory for the UDGs (User Defined Graphics — lines 20-70), and the prompt given for the joystick/keyboard option is POKEd into location 7168 in line 160. To tidy the whole thing up lines 170-200 contain an auto load routine, which will automatically LOAD and RUN the second program. Make sure you SAVE this program before you RUN it, because otherwise the NEW command in line 200 will ruin all your hard work!

To prove to yourself that this works, enter and SAVE Listing 2 immediately after Listing 1. Now RUN Listing 1, and choose your option. Listing 2 will automatically LOAD and RUN and, as you will see, the customised characters and the option chosen, are all intact. You've now got plenty of memory left in both parts for the instructions and the game, and with any luck still some to spare for a few special effects.

```
1 REM LISTING 1
2 :
8 REM RELOCATE MEMORY
9 :
10 POKE52,28:POKE56,28:CLR
17 :
18 REM DEFINE UDGS
19 :
20 FORI=7168TO7679:POKEI,PEEK(I+25600):NEXT
30 FORLC=7448TO7471:READCH:POKELC,CH:NEXT
40 DATA0,4,7,135,120,120,72,108
50 DATA56,68,170,130,68,56,68,198
60 DATA90,36,24,24,255,24,24,102
70 POKE36869,255
77 :
78 REM GIVE OPTION
79 :
80 PRINT"OPTIONS"
90 PRINT"1 KEYBOARD"
100 PRINT"2 JOYSTICK"
110 GETA$:IFVAL(A$)<1ORVAL(A$)>2THEN110
120 IFA$="1"THENPRINT"1 KEYBOARD"
130 IFA$="2"THENPRINT"2 JOYSTICK"
140 PRINT"CHARACTERS"
150 PRINT"# $ %"
157 :
158 REM POKE OPTION INTO LOCATION 7168
159 :
160 POKE7168,VAL(A$)
167 :
168 REM AUTO LOAD
169 :
170 FORI=1TO7:READN:POKE630+I,N:NEXTI
180 POKE198,7
190 DATA76,111,13,82,85,78,13
200 NEW
READY.
```

```
1 REM LISTING 2
2 :
10 POKE36869,255
20 A=PEEK(7168)
30 PRINT"OPTION SELECTED"
40 IFA=1THENPRINT"KEYBOARD"
50 IFA=2THENPRINT"JOYSTICK"
60 PRINT"CHARACTERS"
70 PRINT"# $ %"

READY.
```

A few tricks

In the first part of the program you can afford to go to town on the title screen. The only penalty is that the program will take slightly longer to LOAD. It

can be worth it though. One fairly obvious feature, if you have enough UDGs to spare, is to draw some Hi-Res pictures, or even a personalised signature. This isn't difficult providing you've got the patience to design the graphics. There are however a few simply implemented techniques which can add a bit of sparkle to what otherwise would be fairly boring instructions. For example, instead of simply PRINTing the instructions on the screen one page at a time, why not POKE them on letter by letter, with accompanying sonics. A simple example is shown in Listing 3.

```
1 REM LISTING 3
2 :
3 REM INSTRUCTION POKE
4 :
10 POKE36879,8
20 S=36876:V=36878
30 FORI=1TO6
40 READ A$
50 FORJ=1TOLEN(A$)
60 B$=MID$(A$,J,1):PRINTB$;
70 IFASC(B$)>64THENGOSUB120
80 NEXTJ
90 PRINT
100 NEXTI
110 END
117 :
118 REM SOUND SUBROUTINE
119 :
120 POKEV,15:POKES,180
130 FORT=1TO50:NEXT
140 POKES,0:POKEV,0
150 FORT=1TO50:NEXT
160 RETURN
170 DATA"■■■THIS IS AN EXAMPLE OF"
180 DATA"■■HOW INSTRUCTIONS CAN"
190 DATA"■■BE POKED ON THE SCREEN"
200 DATA"■■■■NOTE THE CURSOR"
210 DATA"■■■■CONTROLS IN THE"
220 DATA"■■■■DATA STATEMENTS"

READY.
```

You can extend this technique to get all sorts of novel effects. Perhaps you could have a plane flying across the screen with the title being pulled along behind a banner? You can have a lot of fun with titles and instructions! Better still why not make use of the memory locations which determine the size and position of the screen. The locations of interest are shown in figure 1.

Location	Description	Normal value	Range
36864	Horizontal Position	12	0 - 22
36865	Vertical Position	38	0 - 155
36866	Width	150	129 - 150
36867	No. Lines	46 (or 74)	Even Nos <46

Figure 1

By playing around with these you can get some really good effects. For example you can scroll the instructions onto the screen smoothly using the technique shown in Listing 4, or if you want something more impressive, you could make each page of instructions disappear down a black hole, to be replaced by the next page. This is demonstrated in Listing 5. If that doesn't appeal, try Listing 6.



You can also make use of these locations in the main game: the ship explodes into fragments with such force that even the screen shakes!

```
1 REM LISTING 4
2 :
3 REM SMOOTH SCROLL
4 :
10 S=36876:V=36878
20 POKE36879,8
30 POKE36865,155
40 PRINT"■"
50 FORI=1TO10
60 PRINT"■■■S-M-O-O-T-H SCROLL"
70 NEXTI
80 FORI=155TO38STEP-1
90 POKE36865,I:GOSUB130
100 NEXTI
110 POKEV,0:POKES,0
120 END
127 :
128 REM SOUND SUBROUTINE
129 :
130 POKEV,15
140 POKES,283-I
150 RETURN

READY.
```

Getting rid of bugs

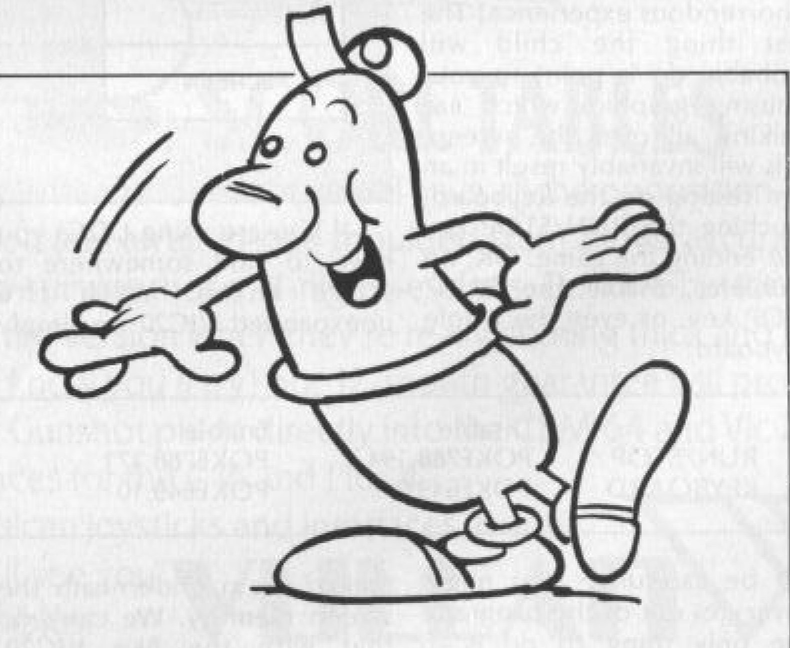
There are two types of bugs. The ones you deliberately design, to be shot down with your laser, and the ones that creep unnoticed into your program. It's the latter that need to be got rid of and that can be tricky. Luckily the VIC 20 is a well tried and tested

```
1 REM LISTING 5
2 :
3 REM BLACK HOLE
4 :
10 POKE36879,24
20 POKE36867,0:POKE36866,129:POKE36865,84:POKE36866,23
30 FORJ=1 TO 3
40 READA$:READCL:POKE36879,CL
50 PRINT"■■■■■■■■■"A$
60 B=0:E=23:S=1:GOSUB200
70 FORT=1TO200:NEXTT
80 B=23:E=0:S=-1:GOSUB200
90 NEXTJ
100 PRINT"■■■■■■■■■■■■END DEMO"
110 POKE36879,24
120 B=0:E=23:S=1:GOSUB200
130 END
200 FORI=BTOESTEPS
210 POKE36867,I*2
220 POKE36866,I+127
230 POKE36865,84-2*I
240 POKE36864,23-INT(I/2)
250 NEXTI
260 RETURN
500 DATA"■FIRST SCREEN",24
510 DATA"■SECOND SCREEN",40
520 DATA"■THIRD SCREEN",104

READY.
```

```
1 REM LISTING 6
2 :
3 REM SICK SCREEN
4 :
10 S=36875:V=36878
20 POKE36879,92
30 PRINT"■■■■■■■■■■■SICK SCREEN"
40 FORI=-8TO8
50 J=ABS(I):GOSUB160
60 POKE36864,I/2+12
70 POKE36865,55-3*J
80 NEXTI
90 FORI=8TO-8STEP-1
100 J=ABS(I):GOSUB160
110 POKE36864,I/2+12
120 POKE36865,55-3*J
130 NEXTI
140 GOTO40
150 GOTO30
160 POKEV,15
170 POKES,(J*2+130)
180 RETURN

READY.
```



machine, and the only bugs you're likely to get are the ones brought about by careless programming. The main problem area occurs when you modify and re-modify programs, and lose all semblance of structure. Even though your program must be squeezed down as tightly as possible it is essential to make sure that the overall structure is intact.

The first stage should be to get a working prototype of your final program. If possible list it out, and then re-write it from scratch, planning each section carefully. It's not difficult at this stage because you should by now know exactly what you're going for. The main thing to avoid is not to jump out of loops or GOSUBs. If you do you'll get those annoying OUT OF MEMORY errors. If you have a replay facility, make sure everything is properly initialised at the start of the program, and if you are using arrays make sure that they are all defined, even if they contain less than 10 elements. Otherwise the VIC will reserve space for 10 elements, and you'll be losing valuable memory.

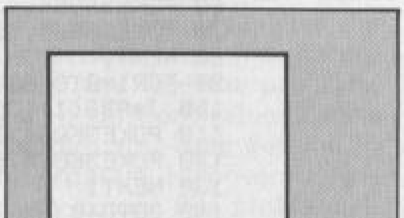
Having re-written the program, you'll probably find it much improved, and you might think that you're home and dry. Not quite! *You* know how the program works, and *you* play the game accordingly. What about the person the game is intended for? If for example the game is intended for a young child, then let one loose on the program. It can be a horrendous experience! The first thing the child will probably do is point to your amusing graphics which are walking all over the screen. This will invariably result in an arm resting on the keyboard, touching the RUN/STOP key and ending the game. OK, no problem, disable the RUN/STOP key, or even the whole keyboard:

	Disable	Enable
RUN/STOP	POKE788,194	POKE788,171
KEYBOARD	POKE649,0	POKE649,10

But be careful — you might never get out of the program! The only thing to do is to persevere. Keep trying ideas, modify the program, and eventually you'll get there. All you need is patience.

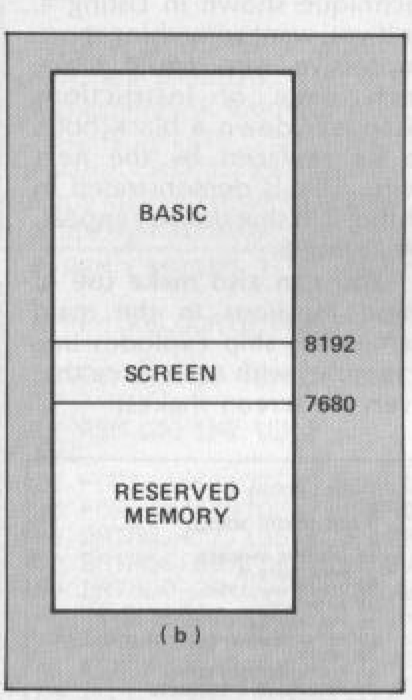
Memory expansion

Up to now I've dealt exclusively with the unexpanded VIC 20. All the techniques and programs described will work equally well on the VIC 20+3K, but problems arise if you want to work with 8K+. This is because the memory map of the VIC 20 is altered with an expansion of 8K and above, as shown in figure 2a. For most arcade type games you don't need expansion, and you're probably better off without it, as it forces tight programming. However, sometimes you will need the extra memory, and you will have to modify your programs accordingly. If you are not using UDGs, it is quite straightforward. All you have to do is remember that the screen character locations start at 4096, and the screen colour locations start at 37888.



(a)

If you are using UDGs you have to find somewhere to store them. With the unexpanded VIC20 we simply tucked them underneath the screen memory. We can't do that with the 8K+ VIC20, because the screen is in a different position. The way round the problem is to relocate the screen, as shown in figure 2b. This can be done either in direct mode, or more elegantly by using a short initialisation program, such as the one shown in Listing 7. In the main program (Listing 8), all you have to do is to set the character pointer to the area of protected memory, starting at 4096, and that's it. You now write the program as normal — even the screen character and colour locations are back in their usual position.



(b)

```
1 :REM LISTING 7
2 :
8 :REM 8K INITIALISATION
9 :
10 POKE43,1:POKE44,32:POKE8192,0
20 POKE641,0:POKE642,32
30 POKE648,30:POKE36866,150
40 POKE36869,240
50 PRINT"J"
57 :
58 REM AUTO LOAD
59 :
60 FORI=1TO7:READN:POKE630+I,N:NEXT
70 POKE198,7
80 DATA76,111,13,82,85,78,13
90 NEW

READY.
```

```
1 REM LISTING 8
2 :
8 REM DEFINE CHARACTER
9 :
10 FORI=4096TO6144:POKEI,PEEK(I+28672):NEXT
20 FORLC=4096TO4103:READCH:POKELC,CH:NEXTLC
30 DATA60,90,60,24,24,36,66,231
40 POKE36869,252
47 :
48 :REM BOTTOM LH CORNER SCREEN CO-ORDS
49 :
50 P1=8164:P2=38884
60 PRINT"J"
67 :
68 :REM POKE CHARACTER ONTO SCREEN
69 :
70 FORX=1TO20STEP2
80 FORY=1TO21STEP2
90 POKEP1+X-22*Y,0:POKEP2+X-22*Y,0
100 NEXTY,X

READY.
```

Some final thoughts

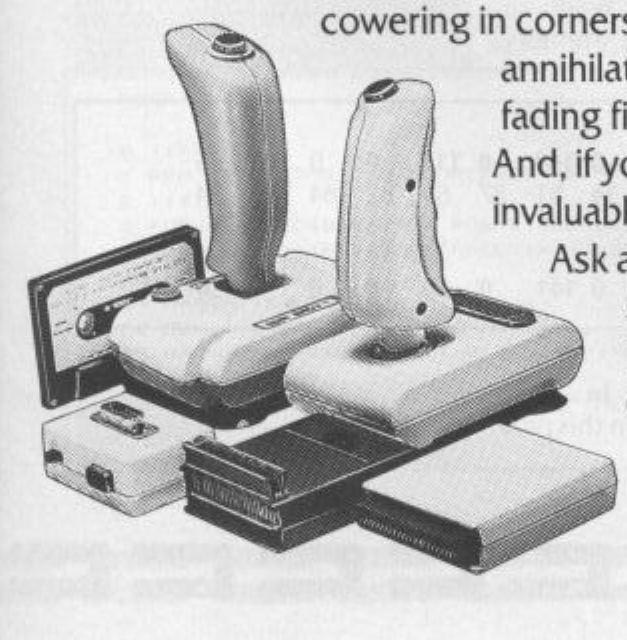
In this series I've outlined some of the techniques which will enable you to write in BASIC. There are lots of other techniques which haven't been covered. Read through listings in magazines, and if you come across something you're not familiar with try it out, and work out how it's done. As a VIC 20 owner you must collect POKEs. I've covered a few in this series, but there are quite a few more, and some of them might be useful!

The most important thing to realise is that by the time you have learnt to write good games you will by necessity have gained a fairly comprehensive knowledge of BASIC, and have a pretty good idea of the way the computer works. When you come to write other programs, such as mathematical analysis, home finance, or educational programs, you will find it comparatively easy. So stick at it. Write those games. It's fun, and at the end of it you'll be quite an expert in BASIC programming.



REWRITE THE HIGH SCORE TABLES

So, you've got a Commodore. You've also got enemies. With the Gunshot, you'll have all the opposition cowering in corners. 8-directional action and an all-in-one moulded stem allows accurate annihilation and strength to survive those all-night sessions. Dual fire buttons for fading fingers (and a rapid fire version when they're really coming thick and fast). And, if you break it (and we know you'll try) our 12-month guarantee will prove invaluable. Only £8.95. The Gunshot plugs directly into the CBM 64 and Vic 20. Ask about Vulcan interfaces for the C 16 and Plus 4.



See the range of Vulcan joysticks and interfaces at your local stockist ... we'll see you on the high score tables.

VULCAN ELECTRONICS LTD

Joystick Genius

200 Brent Street, Hendon, London NW4 1BH

Have you experienced the frustration of losing a program?J.S. Robinson offers a new solution to an old problem.



OLD FOR NEW

HOW DO YOU RECOVER A crashed BASIC program? Where perhaps the computer is Locked Up. You may even have accidently used NEW. The first problem therefore is how to RESET your locked up computer. With both the CBM64 and the VIC20 not having a reset switch you have to improvise.

Resetting

Referring to page 143 of the User Manual or page 360 of the Reference Guide for the CBM64 (VIC20 owners refer to page 283 of Reference Guide) will show details of the User Port. It states grounding of pin 3 will do a cold start, but memory will not be cleared. By momentarily connecting the reset pin 3 to the ground pin 1 with a piece of wire your computer does a cold start. The Serial Port has a reset pin and can be used in the same way.

How BASIC is stored in memory

Having recovered control of your computer you must next consider how a BASIC program is stored in memory.

Enter the following line in direct mode after first switching off and then on again:

```
CBM 64:
FOR X = 2048 TO 2081 : PRINT PEEK (X);: NEXT
VIC 20:
FOR X = 4096 TO 4129 : PRINT PEEK (X);: NEXT
```

You will get on the 64:

0	0	0	88	0	140	0	112	0	0	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	0			

& the VIC20:

0	0	0	88	0	141	0	56	0	0	224	43
74	158	etc.									

The normal BASIC program space starts at 2048 on the CBM64 and 4096 on the VIC20.

Now enter the following short BASIC program:

```
0 :::::
1 PRINT"QWERTY"
2 GOTO1
```

Line 0 should include 5 colons.

Repeat the direct mode entry from above i.e. ForX=2048toetc.. on the 64 or ForX=X4096 on the VIC20. Now memory consists of:

CBM 64:

0	11	8	0	0	58	58	58	58	58	0	25	
8	1	0	153	34	81	87	69	82	84	89	34	
0	32	8	2	0	137	49	0	0	0			

VIC 20:

0	11	16	0	0	58	58	58	58	58	0	25
16	1	0	153	34	81	87	69	82	84	89	34
0	32	16	2	0	137	49	0	0	0		

The contents of address 2049 & 2050 i.e. 11 & 8 are the link address pointing to the beginning of the second line in BASIC i.e. 2059 (8★256+11). The contents of address 2051 & 2052 store the first line no. i.e. 0. Similarly the contents of 4097 & 4098 are the link address pointers on the VIC20.

Now perform a reset or NEW or SYS64738 & memory becomes:

CBM 64:

0	**0**	**0**	**88**	0	**140**	**0**	**112**	**0**	0	0	25
8	1	0	153	34	81	87	69	82	84	89	34

VIC 20:

0	**0**	**0**	**88**	0	**141**	**0**	**56**	**0**	0	0	0

Those date items **in bold** are the only changes in this part of memory.

To recover your listing enter in direct mode again:

CBM 64:
POKE 2049, 11:POKE 2050, 8:POKE2051, 0:POKE2053, 58
POKE2054, 58:POKE2055, 58:POKE2056, 58:POKE2057, 58

VIC 20:
POKE4097, 11:POKE4098, 16:POKE4099, 0:POKE4101, 58:
POKE4102, 58:POKE4103, 58:POKE4104, 58:POKE4105, 58

The program will now LIST; try it. Do not try a RUN, or try to alter the listing at this stage.

It helps to know the first line number. If you are not sure assume it was 0 and you will just corrupt your first line.

The other alterations that occurred in memory when you RESET were to the pointers in address 45 to 50 which are listed below for comparison:

CBM64/VIC20	Pointers to start of BASIC		Pointers to start of arrays		Pointers to end of arrays	
Address:-	45	46	47	48	49	50
Contents on power up	3/3	8/16	3/3	8/16	3/3	8/16
After entry of program	34/34	8/16	41/41	8/16	34/41	8/16
After NEW before LIST	3/3	8/16	10/10	8/16	10/10	8/16

To get your program to run or to be able to alter it in some way address 45 to 50 now have to be calculated & re-established. You won't normally have the benefit of the list above. This is achieved by entering a second program in memory.

Paging to enter a second program.

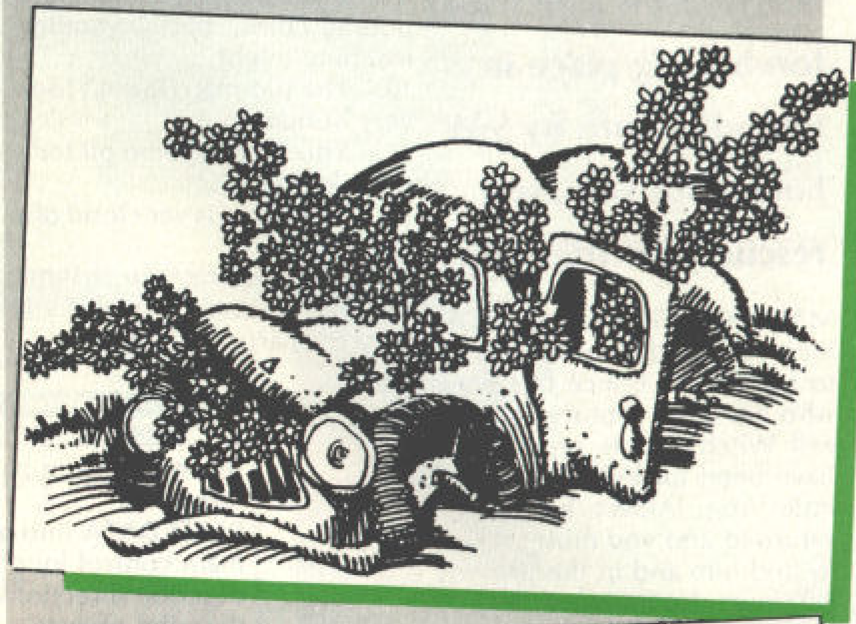
You first need to protect your first program by changing the pointers to start of BASIC:

CBM 64
POKE 43,0 : POKE 44,158 : NEW
VIC 20:
POKE43,1:POKE44,28:POKE7168,0:NEW

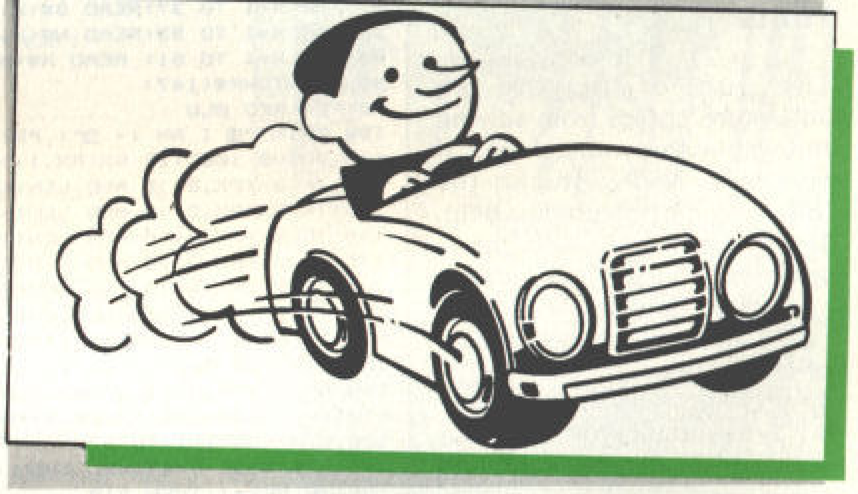
This will now allow you to enter the following short program; SAVE it before you RUN it:

```
READY.

0 :::::
1 REM  RE-ESTABLISH ADDRESSES 45 - 50                         J.S.R.8/10/84
2 :::::
5 FORX=2059TO40960:REM FOR VIC20 USE 4107 TO 7679
10 Z=PEEK(X):IFZ=0THENIFPEEK(X+1)=0THENIFPEEK(X+2)=0THEN20
15 NEXT
20 AX=INT(X/256):AY=X-(AX*256)
24 REM   3*CRD
25 PRINT"■■■P┌45,"AY"+3:P┌46,"AX":P┌47,"AY"+10:P┌48,"AX":P┌49,"AY"+10:P┌50,"AX
29 REM   2*CRD   7*CRU
30 PRINT"■■P┌43,1:P┌44,8■■■■■■■":NEW:REM VIC20 :-POKE43,1:POKE44,16
READY.
```







Hit RETURN over the lines printed by the above program. Your original crashed or newed program will now run. However, you ought first to correct it, if it crashed before. This method of OLDing a BASIC program without resorting to machine code will work with any BASIC program. It has the added advantage of not needing to be in memory prior to a crash or NEW.

FORES

The forest is a dark and foreboding place in this adventure by S W Lucas. But you must rescue your friends...

MANY MONTHS AGO, YOUR lifelong friend, Albert, set out to rescue the Prince of Cenra who has been captured by the evil Witch and is reputed to have been turned into a poor little frog. Albert has never returned and you must set out to find him and in the process you must also find four items of treasure which were stolen from your land by the evil Witch.

This is a traditional adventure game in which you give the computer instructions in the form of two word sentences such as GET LAMP etc. If you want to move north, south, east or west, you only need to enter the first letter of the direction you wish to go in.

Here is a list of some of the words understood:-

GET,TAKE,GRAB,PRAY,HELP, DROP,LEAVE,PULL,KILL, SWIM,SHOOT,ATTACK,N,S, E,W,LOOK,EXAMINE,INVENTORY,USE, THROW,RUB,PUSH

Hints

The fun of playing an adventure comes from solving the game for yourself, but if you are really stuck, the following hints might just help.

1. If you get stuck in a spinning space time vortex, you may as well just give up.
2. Little girls are very fond of chocolates
3. Pawnbrokers like gold!
4. A beanstalk grows better if planted in a good spot and given some food!
5. Medicine will come in useful for a sick person
6. Prayer is usual in sacred places
7. Balrogs don't like anything thrown at them, especially stakes
8. Those mushrooms may be toadstools and I've heard that Orcs are very fond of mushrooms.
9. You will need the right sort of money for the barman, he won't take those new style pound coins, but a vending machine might.
10. The old man doesn't look very human to me!
11. You'll need some oil for a rusty lock
12. The dwarf is very fond of a nip or two
13. Hobgoblins are frightened of dwarves, especially if they are carrying an axe.



Line Explanation

40	sets border and paper colour to white
50-70	titles
75-83	READ DATA into arrays
100-1140	main control loop
100-160	describe directions
170-220	describe objects
230-240	input action
250-1140	call routine for movement scoring etc
1150-2100	routines for get/drop,score,pray etc
2200-	data for game

Variables

P%	current location
S%(x,y)	holds map
B%(x)	pointer to location of objects
N%(x)	pointer to words understood
N$(X)	words recognised
Q$(x)	description of location
G$(x)	description of objects
Z$	input sentence
L$	second word input
B$,C$,D$	substrings of input sentence
L%	pointer to word
R	number of object
A%	score

Program Listing

```
READY.
10 REM ** THE FOREST OF EVIL **
20 REM ** AN ADVENTURE GAME FOR THE COMMODORE 64 **
30 REM ** BY S.W. LUCAS   OCTOBER 1984 **
40 POKE 53280,1:POKE 53281,1
49 REM RED
50 PRINTCHR$(147);TAB(10);"█THE FOREST OF EVIL"
59 REM BLU
60 PRINT:PRINT:PRINT"█AN ADVENTURE GAME FOR THE COMMODORE 64"
69 REM PUR
70 PRINT:PRINT:PRINT:PRINT"█(C)    STEVE W. LUCAS    OCTOBER 1984"
75 DIM Q$(50),G$(50),S%(50,4),V$(5),B%(70),N%(70),N$(70),X$(70)
76 ZS=1:XX%=0:P%=3:M%=0
80 RESTORE:FOR X=1 TO 46:FOR Y=1 TO 4:READ S%(X,Y):NEXT Y:READ Q$(X):NEXT X
81 FOR X=1 TO 37:READ G$(X),B%(X):NEXT X
82 FOR X=1 TO 53:READ N$(X),N%(X):NEXT X
83 FOR X=1 TO 61: READ X$(X):NEXT X
90 PRINTCHR$(147)
99 REM RED BLU
100 PRINT"█ I AM :- █": PRINTQ$(P%):A$=""
110 GOSUB 1000:IF S%(P%,1)>0 THEN A$="NORTH"
120 IF S%(P%,2)>0 AND LEN(A$)>0 THEN A$=A$+",SOUTH"
125 IF S%(P%,2)>0 AND LEN(A$)=0 THEN A$="SOUTH"
130 IF S%(P%,3)>0 AND LEN(A$)>0 THEN A$=A$+",EAST"
135 IF S%(P%,3)>0 AND LEN(A$)=0 THEN A$="EAST"
140 IF S%(P%,4)>0 AND LEN(A$)>0 THEN A$=A$+",WEST"
145 IF S%(P%,4)>0 AND LEN(A$)=0 THEN A$="WEST"
150 IF A%=6 THEN GOTO 11090
159 REM PUR BLU
160 PRINT:PRINT"█ I CAN GO█":PRINTA$
170 IF P%=22 AND S%(22,3)=0 THEN PRINTX$(55)
180 E=0:FOR T=1 TO 37
185 PP%=0:IF B%(T)=P% THEN PP%=1
190 IF PP%=1 THEN 210
200 NEXT T:GOTO 230
209 REM CYN
210 IF E=0 THEN PRINT:PRINT"█THAT LOOKS LIKE :-"
219 REM BLU
220 PRINT"█";G$(T):E=E+1:GOTO 200
229 REM GRN BLU
230 PRINT:PRINT"█WHAT SHALL I DO NOW █";:INPUT Z$
240 B$=LEFT$(Z$,2):C$=LEFT$(Z$,3):D$=LEFT$(Z$,4)
250 PRINTCHR$(147):IF B$="N" AND S%(P%,1)>0 THEN P%=S%(P%,1):GOTO 100
260 IF B$="E" AND S%(P%,3)>0 THEN P%=S%(P%,3):GOTO 100
265 IF B$="S" AND S%(P%,2)>0 THEN P%=S%(P%,2):GOTO 100
270 IF B$="W" AND S%(P%,4)>0 THEN P%=S%(P%,4):GOTO 100
279 REM PUR BLU
280 IF B$="SC" THEN PRINT"█ YOU HAVE SCORED ";A%;" OUT OF 6 █":GOTO 100
290 IF B$="N" OR B$="S" OR B$="E" OR B$="W" THEN PRINT"I CAN'T !":GOTO 100
300 IF B$="PU" THEN 1120
```



ST OF EVIL

Program Listing (cont.)

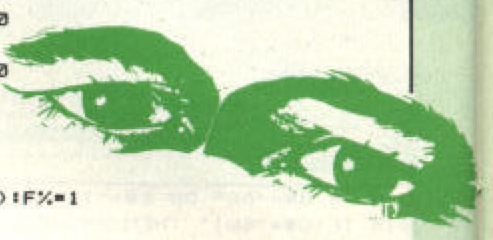
```
310 IF B$="GE" OR B$="TA" OR B$="GR" THEN GOTO 1150
315 IF C$="SWI" THEN PRINT"I'VE NEVER LEARNT TO SWIM!":GOTO 110
320 IF B$="FU" OR B$="PI" THEN PRINT"HOW DARE YOU SAY THAT!":GOTO 100
324 REM BLK
325 IF C$="HEL" THEN PRINT"■I'M AS CONFUSED AS YOU ARE !":GOTO 100
330 IF B$="DR" OR B$="LE" THEN 1860
335 IF C$="LOO" THEN GOTO 100
340 IF C$="PRA" THEN 1300
345 IF C$="SHO" OR C$="HIT" OR C$="ATT" OR C$="KIL"THEN 1340
350 IF C$="EXA" THEN 1370
355 IF C$="THR" OR C$="CHU" THEN 1420
360 IF C$="RUB" THEN PRINT"DON'T BE ABSURD!":GOTO 100
365 IF C$="INV" THEN 1490
370 IF C$="USE" THEN 1440
375 IF C$="QUI" THEN PRINT"YOU QUIT":END
380 PRINT"I'M AFRAID I DON'T UNDERSTAND YOU":GOTO 100
1000 A%=0:IF B%(6)=2 THEN A%=A%+1
1010 IF B%(1)=2 THEN A%=A%+1
1020 IF B%(8)=2 THEN A%=A%+1
1030 IF B%(9)=2 THEN A%=A%+1
1040 IF B%(10)=2 THEN A%=A%+1
1050 IF B%(15)=2 THEN A%=A%+1
1060 RETURN
1075 IF P%=16 AND R=31 THEN PRINTX$(4):GOTO 100
1080 IF P%=20 AND R=52 THEN PRINT"IT'S FASTENED TO A BOX OVERHEAD":GOTO 100
1085 IF P%=27 AND R=32 THEN PRINTX$(5):GOTO 100
1090 IF P%=10 AND R=34 THEN PRINTX$(7):GOTO 100
1095 IF P%=37 AND(R=4 OR R=3) AND XX%<>1 THEN PRINTX$(52):GOTO 100
1100 IF P%=37 AND(R=4 OR R=3) AND XX%=1 THEN PRINT"SHE SMILES AS SHE EATS"
1105 IF P%=42 AND R=33 THEN PRINTX$(6):GOTO 100
1110 IF P%=25 AND(R=35 OR R=36)THEN PRINTX$(8):GOTO 100
1115 IF P%=14 AND(R=1 OR R=2)AND B%(2)=-1 THEN1810
1120 IF P%=20 AND R=52 THEN PRINT"I FLUSH THE TOILET!"
1130 IF P%<>20 THEN PRINT"PULL WHAT ?"
1140 GOTO 100
1150 GOSUB 2000:IF L$="PLANT" THEN PRINT"SORRY!":GOTO 100
1155 IF L%<>1 THEN GOTO 100
1160 E%=0:FOR H=1 TO 37 : IF B%(H)=P% AND B%(N%(R))=P% THEN E%=1
1165 NEXT:IF E%=0 THEN GOTO 100
1170 IF P%=12 AND R=30 THEN PRINTX$(3):GOTO 100
1175 IF P%=45 AND (R=37 OR R=38) THEN PRINTX$(45):GOTO 100
1180 IF P%=34 AND R=39 THEN PRINTX$(46):GOTO 100
1185 IF P%=21 AND (R=44 OR R=45) THEN PRINTX$(49):GOTO 100
1190 IF P%=14 AND (R=1 OR R=2) AND B%(1)<>14 THEN PRINTX$(17):GOTO 100
1195 IF P%=13 AND R=19 AND B%(13)<>13 THEN PRINTX$(19):GOTO 100
1200 IF P%=13 AND R=18 AND B%(13)<>13 THEN PRINTX$(10):GOTO 100
1205 IF P%=9 AND (R=6 OR R=7) AND B%(3)=9 THEN 1810
1210 IF P%=9 AND (R=6 OR R=7) AND B%(13)=9 THEN PRINTX$(21):GOTO 100
1215 IF P%=9 AND (R=6 OR R=7) AND B%(13)<>9 THEN PRINTX$(36):GOTO 100
1220 IF P%=29 AND B%(4)<>29 THEN PRINTX$(54):GOTO 100
1225 IF P%=28 AND (R=15 OR R=16) AND B%(11)<>28 THEN PRINTX$(56):GOTO 100
1230 IF P%=28 AND (R=15 OR R=16) AND B%(11)=28 THEN G$(10)=X$(57):B%(11)=-1
1240 GOTO 1810
1299 REM BLK
1300 IF P%<>27 THEN PRINT"■THAT FEELS BETTER...BUT NOTHING HAPPENS":GOTO 100
1310 S%(27,2)=39:S%(27,3)=28:G$(22)=X$(58)
1319 REM PUR
1320 PRINT"■THAT'S ABSOLUTELY AMAZING. THE ALTAR    HAS SLID ASIDE":GOTO 100
1340 IF P%=10 AND B%(7)=0 THEN PRINTX$(27):S%(10,3)=5:G$(24)=X$(28):GOTO 100
1345 IF P%=29 AND M%=0 THEN PRINTX$(29):GOTO 100
1350 IF B%(7)<>0 THEN PRINT"I DON'T HAVE ANY WEAPON!":GOTO 100
1355 PRINTX$(30):GOTO 100
1370 GOSUB 2000:IFB%(8)=0ANDR=12 THEN PRINTX$(34):GOTO100
1375 IFP%=12AND(R=27ORR=26)THENPRINTX$(40):GOTO 100
1380 IF P%=25AND(R=35ORR=36)THEN PRINTX$(59)
1385 IFR=15ANDB%(15)=0THENPRINTX$(22):GOTO100
1390 IF P%=24 THEN PRINTX$(37):GOTO 100
1395 IF P%=39ANDR=21THENPRINTX$(41):GOTO100
1400 IF(R=1ORR=2)ANDP%=14 THEN PRINTX$(42):GOTO 100
1405 IF(R=48ORR=47)ANDB%(33)=0 THEN PRINTX$(23):GOTO 100
1410 IF(R=22ORR=23)ANDB%(15)=0 THEN PRINTX$(22):GOTO 100
1415 PRINTX$(35):GOTO 100
```







Program Listing (cont.)



```
1420 IFP%=16 AND B%(21)=16 THEN PRINTX$(48):S%(16,2)=21:S%(16,3)=22
1425 IFP%=16 AND B%(21)=16 THEN B%(21)=0:GOTO 100
1430 GOTO 100
1440 GOSUB 2000:IF P%<>42 THEN PRINTX$(32):GOTO 100
1445 IF R<>28 THEN PRINTX$(32):GOTO 100
1460 PRINTX$(31):S%(42,3)=40:G$(23)=X$(24):GOTO 100
1489 REM PUR
1490 PRINT"■I AM CARRYING ;- "
1494 REM GRN RED
1495 PRINT"■A CRUCIFIX AROUND MY NECK AND■"
1500 F%=0:FOR H=1 TO 3:IF V$(H)<>"" THEN PRINTV$(H):F%=1
1505 NEXT:IF F%=0 THEN PRINT"NOTHING ELSE"
1510 GOTO 100
1810 E%=0:FOR D=1 TO 3:IF V$(D)="" THEN V$(D)=G$(N%(R)):E%=1:D=5
1820 NEXT:IF E%=0 THEN PRINTX$(38):GOTO 100
1830 B%(N%(R))=0:GOTO 100
1859 REM BLK
1860 GOSUB 2000:IF L%<>1 THEN PRINT"■I CAN'T SEE A ";L$:GOTO 100
1870 E%=0:FOR X=1 TO 3:IF V$(X)=G$(N%(R)) THEN V$(X)="":E%=1
1880 NEXT:IF E%<>1 THEN PRINTX$(39):GOTO 100
1890 B%(N%(R))=P%
1900 IFP%=7ANDR=24THENPRINTX$(1):S%(7,2)=12:B%(16)=0:Q$(7)="BY AN OPEN GATE"
1905 IF P%=12 AND(R=26 OR R=27)THENPRINTX$(2):S%(12,2)=17:S%(12,3)=13
1910 IF P%=12 AND(R=26 OR R=27)THEN B%(20)=-1:B%(18)=0
1915 IF P%=14AND(R=4ORR=3)THENPRINTX$(9):G$(1)=X$(13):B%(2)=-1:GOTO 100
1920 IF P%=37 AND (R=18 OR R=19) THEN XX%=1
1925 IF P%=29AND(R=7 OR R=6) THEN G$(5)=X$(14):PRINTX$(12):GOTO 100
1930 IFP%=28ANDR=17 THENPRINTX$(16):G$(10)="MY FRIEND ALBERT":GOTO 100
1935 IFP%=13ANDR=20THENPRINTX$(18):G$(12)="A BOX OF CHOCOLATES":GOTO 100
1940 IF P%=45 AND R=25 THEN PRINTX$(20):S%(45,1)=46:B%(26)=-1:B%(17)=-1:GOTO100
1945 IFP%=22AND(R=50ORR=49)ANDB%(33)=22THENPRINTX$(53):S%(22,2)=30:S%(22,3)=23
1950 IFP%=22AND(R=50ORR=49)ANDB%(33)=22THEN G$(33)=X$(61):X$(55)="":GOTO 100
1955 IF P%=22 AND (R=47ORR=48)AND B%(34)=22THENPRINTX$(53):S%(22,2)=30:S%(22,3)
=23
1960 IFP%=22AND(R=47ORR=48)ANDB%(34)=22THENG$(33)=X$(61):X$(55)="":GOTO100
1965 IF P%=27 AND B%(12)=27 AND B%(1)=0 THENS%(27,3)=28:B%(12)=0:G$(22)=X$(58)
1970 IFP%=21ANDR=53THENS%(21,2)=29:PRINTX$(47):B%(32)=0:B%(53)=0:GOTO 100
1975 IFP%=13AND(R=19ORR=18)ANDB%(13)=13THENPRINTX$(18):GOTO100
1980 IFP%=34ANDR=8THENPRINTX$(25):S%(34,3)=35:B%(5)=-1:B%(27)=-1:GOTO 100
1995 GOTO 100
2000 L$="":FOR H=1 TO LEN(Z$)
2010 IF MID$(Z$,H,1)=" " THEN L$=RIGHT$(Z$,(LEN(Z$)-H)):H=90
2020 NEXT:R=0:L%=0
2030 IF LEN(L$)<2 THEN RETURN
2040 FOR H=1 TO 53:IF LEFT$(N$(H),LEN(L$))=L$ THEN L%=1:R=H
2050 NEXT:RETURN
2100 GOTO 100
10000 DATA 0,0,2,0,OUTSIDE AN OLD HUT. THE DOOR IS LOCKED.
10010 DATA 43,8,3,1,IN A PAWNBROKER'S SHOP. A SIGN READS    DROP TREASURES HERE
10020 DATA 0,9,4,2,WALKING ALONG A DISUSED RAILWAY LINE
10030 DATA 0,10,0,3,ON A FOOTPATH LEADING INTO THE FOREST
10040 DATA 0,11,6,10,AT A JUNCTION IN THE FOOTPATH
10050 DATA 0,0,0,5,IN A QUARRY. I CAN GO NO FURTHER
10060 DATA 0,0,8,0,BY A GATE. A GREEDY PAWNBROKER REFUSES  TO LET ME PASS
10070 DATA 2,0,0,7,IN A YARD FULL OF DUSTBINS
10080 DATA 3,0,0,0,IN THE 'OLD SHIP' INN
10090 DATA 4,0,0,0,IN A GLOOMY WOOD.
10100 DATA 5,0,16,0,IN A SMALL THICKET
10110 DATA 7,0,0,0,IN A CLEARING FULL OF BROKEN POTS. IT'S OVERGROWN WITH WEEDS
10120 DATA 0,0,14,12,BY A SMALL PLOT OF LAND. IT'S VERY      OVERGROWN.
10130 DATA 0,18,0,13,BY A GRAVESTONE. IT'S ENGRAVED 'COUNT   DRACULA'
10140 DATA 0,19,0,0,IN A SMALL OUTBUILDING.
10150 DATA 0,0,0,11,AT A DEAD END.
10160 DATA 12,25,26,0,IN A DARK TUNNEL WITH WATER DRIPPING    FROM ABOVE
10170 DATA 14,0,19,0,ON A MOUNTAIN TRACK.
10180 DATA 15,27,0,18,BY THE ENTRANCE TO A TEMPLE.
10190 DATA 0,28,0,0,IN A LABORATORY FULL OF STRANGE         APPARATUS
10200 DATA 16,0,0,0,ON A NARROW BRIDGE OVER A FAST FLOWING  RIVER.
10210 DATA 0,0,0,16,IN A DARK CLEARING IN THE MIDDLE OF THE FOREST.
10220 DATA 34,0,0,22,IN A SMALL WORKSHOP.
10230 DATA 0,0,31,0,IN A VAST HALL LIT BY CANDLELIGHT.
10240 DATA 17,0,42,0,IN A NARROW TWISTY PASSAGE.
10250 DATA 0,42,0,17,IN A LARGE HALL FULL OF WIERD STATUES.
10260 DATA 19,0,0,0,IN A TEMPLE TO THE SUN GOD.
10270 DATA 20,38,0,27,IN A HOSPITAL WARD
10280 DATA 21,0,0,0,AT A DEAD END.
10290 DATA 22,0,32,0,IN A SMALL SHRINE. A CANDLE BURNS       BRIGHTLY.
10300 DATA 32,0,0,24,ON A SANDY BEACH. A NOTICE READS 'NO    SWIMMING HERE'
10310 DATA 0,31,33,30,IN A LARGE ARENA. A MAN LEANS AGAINST   THE WALL.
10320 DATA 36,37,0,32,IN A SMALL COTTAGE.
10330 DATA 0,23,0,0,IN A SMALL STOREROOM.
```

Program Listing (cont.)

```
10340 DATA 41,41,41,34,IN A SPINNING VORTEX IN SPACE.. I DON'T UNDERSTAND IT!
10350 DATA 36,36,36,36,SPINNING IN SPACE AND TIME!
10360 DATA 33,0,0,0,IN A PRISON CELL.
10370 DATA 28,0,0,0,IN A BROOM CUPBOARD. IT'S FULL OF SHOES.
10380 DATA 27,0,0,0,IN A LONG NARROW TUNNEL. I SEE DAYLIGHT IN THE DISTANCE
10390 DATA 0,0,0,42,IN A CRYPT. I HEAR STRANGE NOISES!
10400 DATA 41,41,41,35,IN A DENSE FOG.
10410 DATA 26,0,0,25,IN A SMALL CAVERN. IT DOESN'T LOOK      NATURAL!
10420 DATA 0,2,45,44,IN A LARGE GLASSHOUSE.
10430 DATA 0,0,43,0,AT THE FAR END OF THE GLASSHOUSE.
10440 DATA 0,0,0,43,OUTSIDE A SMALL POTTING SHED.
10450 DATA 0,45,0,0,IN A SMALL POTTING SHED.
10460 DATA A LARGE FROG,14,A LITTLE GIRL,37,A WALLET FULL OF MONEY,30
10470 DATA A FLASK OF BRANDY,9,AN OLD DWARF,29,A ** RUBY** ENCRUSTED GOBLET,33
10480 DATA A LOADED RIFLE,46,A ** DIAMOND ** TIARA,41
10490 DATA AN ANTIQUE ORC ** DAGGER **,38,MY OLD FRIEND ** ALBERT **,28
10500 DATA A JAR OF STRANGE MEDICINE,31,A VENDING MACHINE FULL OF CHOCOLATES,13
10510 DATA A POUND COIN,11,A BAUBLE,39,A ** SILVER ** SPEAR,40
10520 DATA A GOLD PLATED RING,4,AN OIL CAN,15,A LARGE PLANT POT,43
10530 DATA A GENUINE PLASTIC CRUCIFIX,6
10540 DATA TWO FUNNY LITTLE MEN CALLED BILL & BEN WHO GUARD A SMALL ENTRANCE,12
10550 DATA AN EVIL BALROG,16,A LARGE ALTAR,27,A MENACING VAMPIRE,42
10560 DATA A SEEMINGLY INNOCENT OLD MAN WHO REFUSES TO LET ME PAST,10
10570 DATA A COFFIN COVERED IN MUD,25,A RUSTY LOCK,45
10580 DATA A DANGEROUS HOBGOBLIN WITH HALITOSIS,34,A CASKET WITH NO LID,26
10590 DATA A BROKEN LID,24,A SMALL HORSE,32,A SLEEPING MONSTER,23
10600 DATA A GREEDY ORC WARRIOR GUARDING THE BRIDGE,21
10610 DATA A TINY BEANSTALK WITH A LABEL ATTACHED,3,A PACKET OF FERTILISER,44
10620 DATA A SMALL BRASS ORNAMENT,2,A CHAIN,20,SOME MUSHROOMS,18
10630 DATA FROG,1,PRINCE,1,GIRL,2,LITTLE,2,WALLET,3,FLASK,4,BRANDY,4
10640 DATA DWARF,5,RUBY,6,RIFLE,7,GUN,7,DIAMOND,8,ANTIQUE,9,DAGGER,9,ALBERT,10
10650 DATA FRIEND,10,MEDICINE,11,VENDING,12,CHOCOLATES,12,COIN,13,BAUBLE,14
10660 DATA GOLDEN,15,SPEAR,15,RING,16,OIL,17,PLANT POT,18,POT,18
10665 DATA CRUCIFIX,19,CROSS,19,GUARD,20
10670 DATA BALROG,21,ALTAR,22,VAMPIRE,23,MAN,24,COFFIN,25,CORPSE,25,DOOR,26
10680 DATA LOCK,26,GOBLIN,27,CASKET,28,LID,29,HORSE,30,SLEEPING,31,MONSTER,31
10690 DATA ORC,32,WARRIOR,32,BEANSTALK,33,LABEL,33,PACKET,34,FERTILISER,34
10700 DATA MONKEY,35,CHAIN,36,MUSHROOMS,37
10710 DATA THE PAWNBROKER RUBS HIS HANDS IN GLEE AS HE GRABS IT AND RUNS IN
10720 DATA BEN GRABS IT AND RUNS AWAY SHOUTING      SOMETHING ABOUT A PRESENT
10730 DATA YOU MUST BE JOKING,HAVE YOU SEEN THOSE FANGS ?
10740 DATA I CANT'T LIFT IT!,WHAT DO YOU THINK I AM ?
10750 DATA ALL THAT GLITTERS IS NOT GOLD!,I'M NOT CARRYING THAT !
10760 DATA THE GIRL CHASES THE FROG AND KISSES IT. IT TURNS INTO A PRINCE
10770 DATA I'M NOT A THIEF YOU KNOW. I NEED A COIN.
10780 DATA O.K.,THE OLD DWARF DRINKS THE BRANDY AND     SMILES
10790 DATA A HANDSOME PRINCE,AN OLD DWARF. HE'S HAPPY TO COME ALONG  WITH ME.
10800 DATA ALBERT ISN'T FEELING WELL ENOUGH TO COME,HEALTHY ALBERT
10810 DATA THE FROG HOPS AWAY,THE COIN FITS AND I GET A BAR OF CHOCS !
10820 DATA I CAN'T CARRY THE WHOLE MACHINE!,THE LOCK OPENS !
10830 DATA THE BARMAN GRINS AND SAYS 'I DON'T      ACCEPT THOSE'
10840 DATA THERE'S AN INSCRIPTION ON IT,IT READS   'USE GROWMORE FERTILISER'
10850 DATA THE DARK OPENING TO A CRYPT,THE DWARF CHASES IT AWAY WITH HIS AXE
10860 DATA SEEMS LIKE A GOOD IDEA BUT NOTHING HAPPENS!
10870 DATA THE BODY SHRIVELS AWAY AND A PILE OF    DUST FALLS TO THE GROUND
10880 DATA AN EVIL SMELLING MONSTER WITH BLUE VENOM DRIPPING FROM ITS FANGS
10890 DATA WHOOPS MISSED !,BANG.  NOTHING HAPPENED,THE VAMPIRE VANISHES
10900 DATA IT DOESN'T SEEM TO DO ANYTHING AT THE   MOMENT
10910 DATA THE VAMPIRE'S ALREADY GONE!,THE DIAMONDS MUST BE VALUABLE
10920 DATA I CAN'T SEE ANYTHING UNUSUAL,THE BARMAN DOESN'T LIKE THAT!
10930 DATA IT BEARS THE INSCRIPTION 'A PERFECT GIFT FOR ALL'
10940 DATA MY HANDS ARE FULL!,I'VE NOT GOT IT DUMMY!
10950 DATA THE POTS ARE VERY DIRTY!,A WORTHLESS BAUBLE!,IT'S TOO FAST FOR ME
10960 DATA OH DEAR. I THINK I'VE BEEN SPOTTED,I'VE JUST BEEN ATTACKED
10970 DATA IT'S TOO RUSTY TO MOVE,YOU MUST BE JOKING!
10980 DATA IT EATS THE MUSHROOM AND GOES INTO A STUPOR
10990 DATA STRAIGHT THROUGH THE HEART. I GRAB THE  STAKE BACK QUICKLY
11000 DATA ORCS ARE FAR TOO DANGEROUS!,SEEMS LIKE A GOOD IDEA
11010 DATA THE HORSE HAS JUST FAINTED,THE GIRL REFUSES TO COME
11020 DATA THE BEAN SPURTS INTO RAPID GROWTH. I CAN USE IT TO CLIMB OVER
11030 DATA THE DWARF REFUSES
11040 DATA THERE'S A WALL HERE BUT I CAN'T CLIMB IT. THE SOIL IS FERTILE HERE.
11050 DATA ALBERT IS TOO ILL TO TRAVEL,MY HEALTHY FRIEND ** ALBERT **
11060 DATA THE ALTAR HAS MOVED ASIDE AND THERE ARE TWO PASSAGES BEHIND IT
11070 DATA THERE'S A HALF DECOMPOSED BODY IN THERE,I'M SORRY I DON'T UNDERSTAND
11080 DATA THE BEAN GROWS SO MUCH THAT I CAN CLIMB THE WALL
11089 REM RED
11090 PRINT CHR$(147):PRINTTAB(10)"█W E L L  D O N E"
11099 REM BLU
11100 PRINT:PRINT:PRINT:PRINT"█YOU HAVE RESCUED THE PRINCE,"
11110 PRINT"FOUND ALBERT AND RECOVERED THE TREASURE"
READY.
```



Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

SO YOU OWN A COMMODORE?

SO YOU'VE WRITTEN SOME PROGRAMS?

SO WHY HAVEN'T YOU SUBMITTED THEM TO US?

Your Commodore is always on the lookout for new material for publication and we know that there are thousands of intelligent, literate, innovative and creative Commodore owners out there, so why don't we get together?

If you have written an exhilirating game or an invaluable utility on your Commodore micro, share your talents with us and our readers by submitting your efforts and the form to the address below. All articles should be documented and type-written and should be accompanied by a printout of the program as well as a copy of the program on cassette or disc. All material should be original; if it is not chosen for publication, it will be returned to you.

You may not have written any software yourself, but you have very firm opinions about the world of Commodore and all their attendant industries and products. Then put your opinions on paper and post them to us, again at the address below — you never know, you might even get paid for airing your views! All submissions should be sent to:

The Editor
Your Commodore
Argus Specialist Publications Limited
No 1 Golden Square
London WIR 3AB

★PLEASE COMPLETE IN BLOCK CAPITALS

Your Name

Program Name

Computer/memory size it runs on

Amount of memory program occupies

Other computers/memory size which your program runs on without conversion or use

Does your game need or use joysticks? Yes No

Have you sent your game to another magazine Yes No

Is it original/or a variation on a theme?

Your Address

Telephone Number

Times to contact you

VIZASTAR 64

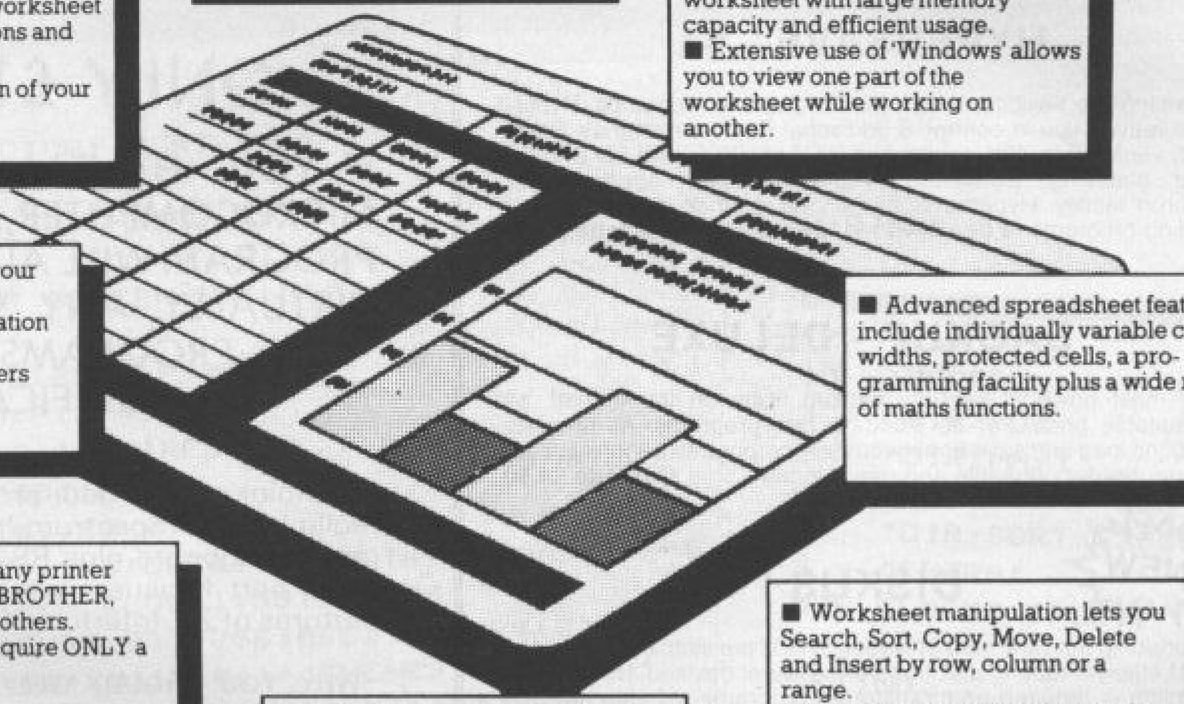
THE INFORMATION PROCESSOR

■ Spreadsheet ■ Database ■ Graphics

■ One word menu lines for easy selection of facilities.
■ The high resolution grid pattern frames the worksheet giving an easy-to read display.

■ Extract from your information files and include them into the worksheet for fast, powerful calculations and easy reporting.
■ Instant, on-screen design of your information records.

■ A 1000 Row by 64 Column wide worksheet with large memory capacity and efficient usage.
■ Extensive use of 'Windows' allows you to view one part of the worksheet while working on another.



■ Add or remove items from your records without having to re-construct your existing information records.
■ Records up to 1,000 characters long with each item up to 250 characters.

■ Advanced spreadsheet features include individually variable column widths, protected cells, a programming facility plus a wide range of maths functions.

■ Supports virtually any printer –CBM, EPSON, JUKI, BROTHER, OLIVETTI and many others.
■ Parallel printers require ONLY a low-cost cable.

■ Worksheet manipulation lets you Search, Sort, Copy, Move, Delete and Insert by row, column or a range.
■ Text editing facilities for letters, invoices, report headings etc.

■ Include information directly into the Worksheet from VizaWrite and almost any other system.
■ Simultaneous display of worksheet and line or bar graphs.

VizaStar integrates three important electronic aids to your business or home office.
A fast and easy-to-use package – VizaStar is a comprehensive information processor that includes an electronic worksheet, information filing system and simultaneous on-screen graphics.

VizaStar has been designed to incorporate the very latest in user-friendliness – so now you can handle your all-important information in many new ways and in a quick and effective manner.

The information filing capabilities of VizaStar enable you to store your information onto disk in an ordered and instantly retrievable form. This means that you can quickly pick out individual or entire groups of related information to create lists, statements, labels or reports.

VizaStar is the ONLY program of its kind on the Commodore 64. Completely consistent with the VizaWrite word processor, VizaStar provides a totally comprehensive office system.

VizaStar costs just £99.95 (incl. VAT) from your computer dealer and selected stores. Please write for more details and the name and address of your nearest dealer.

AVAILABLE NOW - VIZASTAR 64
£99.95 (inc VAT)

VIZA SOFTWARE

VIZA SOFTWARE, 9 MANSION ROW, BROMPTON, GILLINGHAM, KENT ME7 5SE
TEL: MEDWAY (0634) 813780
Dealer enquiries welcome.

COMMODORE 64

FAST LOADERS   TAPE TO DISK

R. B. S.
(RAPID BACKUP SYSTEM)

A NEW GENERATION conversion utility, developed from the acclaimed Hypersave-64, R.B.S. converts virtually all your long loading cassette programs to TURBO-LOAD, faster than the CBM disk drive. Multi-part and autorun programs are handled with ease. Converted programs load independently. No additional hardware, no Pokes, no SYS calls, no user knowledge required. Load "The Hobbit" in 120 seconds. Tornado, Lightning load, Fastback now obsolete. R.B.S. will convert more programs than any competing utility. Cassette £7.50p

HYPERSAVE - 64

Retaining the flexibility which the Programmer demands, Hypersave leaves you in control. 8 additional Basic commands. Save, load, verify up to 48K continuous RAM at HYPERSPEED with full error checking. Converted Programs reload independently. Autorun facility. Hypersave also allows a small number of fast loading programs to be converted to Hyperload. **Cassette £7.50**

BACKUP - DELUXE

The most powerful security backup utility on the market, an unbeatable price. For all standard rate programs. All program sections load and save continuously. Incorporates a sophisticated header reader. Versatile and user friendly. **Cassette £4.50**

NEW

DISKUS I

Introducing the new market leader. This is the simplest to use and most efficient tape to disk transfer utility yet devised. No program rewriting is required and transferred programs will automatically run when loaded. *For single, multi, and autorun programs. *Automatic filename handling. *Efficient use of Disk space. * No user knowledge required. Diskus 1 will handle several proram types which other utilities ignore, including headerless files. A bonus program is included which will transfer a number of fast loading programs to disk. **Cassette £9.99**

PRO - SPRITE

Library, Editor, Animator for single and multicolour sprites. Joystick control. Packed with features. Menu driven. 100% machine code. Complete with your first sprite library. **Cassette £7.50**

DISKUS 1 is **£9.99 SPECIAL DISCOUNT** any two of the remainder just **£9.99**. All four only **£19.99**. Free secrets! I/O guide with every order.

Fast despatch guaranteed. Cheque/P.O. or S.A.E. to:

DOSOFT
2 Oakmoor Avenue, Blackpool FY2 0EE.

ZX* SPECTRUM

SIMULATOR FOR THE commodore 64+

ONLY £14 - 95 !

IF YOU HAVE A COMMODORE-64 MICROCOMPUTER, THIS BRILLIANT PROGRAM WILL ALLOW YOU TO USE VIRTUALLY ANY SPECTRUM BASIC PROGRAMS WITHOUT MODIFICATION !

Just think of it ! - your programmes will think you really have a Spectrum ! Because your 64 has a proper keyboard plus RS232, disk-handling, and serial port facilities, we are throwing in a lot of features of ZX-Interface-1 * as well.

WHY YOU SHOULD GET A SPECTRUM SIMULATOR

! You can type in (virtually) all these Spectrum listings from magazines, as well as those for the 64.

! It's an incredibly cheap way to get all the facilities of another computer.

! And if you're upgrading from Spectrum to 64, your old programs don't all become obsolete.

! Spectrum Basic is in many ways easier and more powerful than Commodore Basic. Why not compare them for yourself ?

FULL TECHNICAL DETAILS AVAILABLE FREE

* Trademarks of Sinclair Research Ltd
\+ Trademark of Commodore Business Machines Ltd.

ZX-SPECTRUM SIMULATOR

FOR SPEEDY MAIL ORDER SERVICE, SEND CHEQUE/P.O. FOR £14.95 (Access/Visa accepted) to

WHITBY COMPUTERS LTD

7. CHUBB HILL ROAD, WHITBY, N.YORKS, YO21 1JU
Tel : 0947-604966

Do you own a disc drive? If so use this program from M. C. Hart to sort your directory.

DIRECTORY SORT

THIS PROGRAM READS THE directory from the disc and then prints it out 'straight' and also after an alphabetical sort. The program was developed on a Commodore 64 with a 1541 disc drive but ought to work on almost any Commodore machine. I have also tested it on a PET 4032 plus 3040 disc drive and it worked all right. I cannot vouch for the 8050 disc drives which have different internal lay-outs.

The program is designed as a sub-routine which can be appended to programs already in memory. For a Commodore 64 and with a program (with lower line numbers!) already in memory, follow this procedure:

(a) POKE 43,PEEK (45)–2: POKE 44,PEEK(46)
(b) LOAD "DIRECTORY SORT", device no.
(c) POKE43,1: POKE 44,8 and off you go with a RUN 63000.

Program Listing

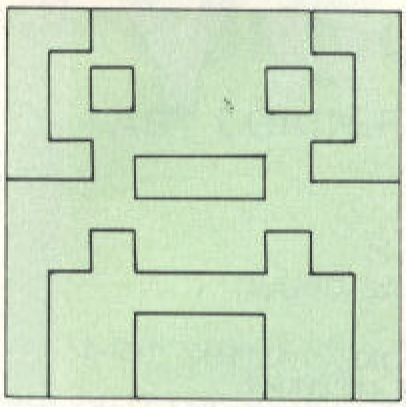
```
63000 :
63010 REM *** DIRECTORY SORT ***
63020 REM
63030 REM *** BY M. C. HART  ***
63040 REM
63050 REM ** OUTPUT SCREEN/PRINTER **
63060 REM
63070 DIM F$(250),F(250),P(250)
63080 FOR J=1TO250:P(J)=J:NEXT
63090 PRINT"READING IN DIRECTORY...":PRINT
63100 OPEN 1,8,0,"$0":N=0
63110 GET#1,X$:GET#1,X$
63120 GET#1,X$:GET#1,X$:GET#1,Y$:GET#1,A$
63130 IFST OR DS THEN CLOSE1:GOTO63200
63140 F(N)=ASC(X$+CHR$(0))+256*ASC(Y$+CHR$(0))
63150 GET#1,X$:A$=A$+X$:IFX$<>""THEN63150
63160 C=0:FORK=1TO4:IFASC(MID$(A$,K))=34THENC=K
63170 NEXT:IFC<>0THENA$=MID$(A$,C)
63180 F$(N)=A$:A$="":N=N+1:GOTO63120
63190 :
63200 REM OUTPUT
63210 INPUT"OUTPUT TO SCREEN OR PRINTER (S/P)";Q$
63220 IF Q$="S"THEN DN=3:GOTO63250
63230 IF Q$="P"THEN DN=4:GOTO63250
63240 GOTO 63210
63250 OPENDN,DN:PRINT#DN:PRINT#DN,"ORIGINAL..."
63260 :
63270 PRINT#DN:PRINT#DN,F(0);CHR$(18);F$(0)
63280 N=N-1:FORJ=1TON:PRINT#DN,F(J);SPC(4-LEN(STR$(F(J))));F$(J):NEXT
63290 :
63300 PRINT#DN:PRINT#DN,"SORTING..."
63310 M=INT(N/2):FORA=-1TO0:FORJ=1TON-M:FORH=JTO1STEP-M:L=H+M
63320 K=0:IFF$(P(H))>F$(P(L))THENZ=P(H):P(H)=P(L):P(L)=Z:K=H
63330 H=K:NEXT:NEXT:M=INT(M/2):A=(M>0):NEXT
63340 :
63350 PRINT#DN:PRINT#DN,F(0);CHR$(18);F$(0)
63360 FORJ=1TON:PRINT#DN,F(P(J));SPC(4-LEN(STR$(F(P(J)))));F$(P(J)):NEXT
63370 CLOSE DN
READY.
```

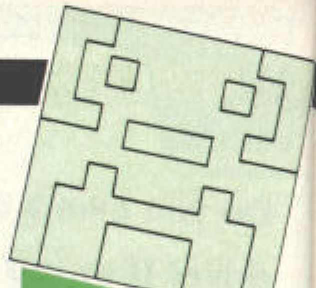
```
ORIGINAL...

0
42   "B.AID"                PRG
10   "SUPERMON"             PRG
3    "DIR/PG"               PRG
2    "DIR/MH/1"             PRG
4    "DIRECTORY SORT"       PRG
4    "DIR-SORT/63000"       PRG
2    "EXTRA"                PRG
596 BLOCKS FREE.

SORTING...

0
42   "B.AID"                PRG
4    "DIR-SORT/63000"       PRG
2    "DIR/MH/1"             PRG
3    "DIR/PG"               PRG
4    "DIRECTORY SORT"       PRG
2    "EXTRA"                PRG
10   "SUPERMON"             PRG
596 BLOCKS FREE.
```





GREEN THINGS FROM OUTER SPACE

An epic space adventure from Mike Roberts in which the safety of earth itself is at stake!

GREEN THINGS FROM OUTER space is a game of extreme skill and mind bending strategy — play it if you dare!

Green Things from Outer Space runs on a C16 and will run on a Plus/4 with no modification at all.

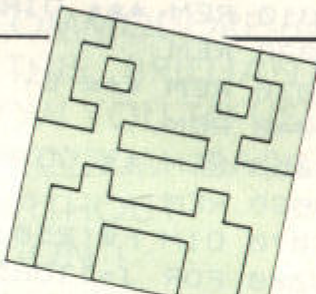
The object of the game is to stop the green things getting to the bottom of the screen. There are two green things on the screen at any one time and your laser base at the bottom. The screen is 24 shades of blue — going from dark at the top (near space) to a bright blue at the bottom. Your ship is moved left and right by the Z and X keys, the laser blast is fired by the ? key.

The program contains UDGs in text mode to increase speed and reduce memory consumption. There are no commands to access these in the very comprehensive BASIC that the C16 and Plus/4 comes with. So I have had to use POKEs. Doing this on a C16 is very awkward as you have to look after the memory banking and set pointers and the area of memory. Astute programmers will understand what I am doing; everyone else will have to wait until the techniques become better known.

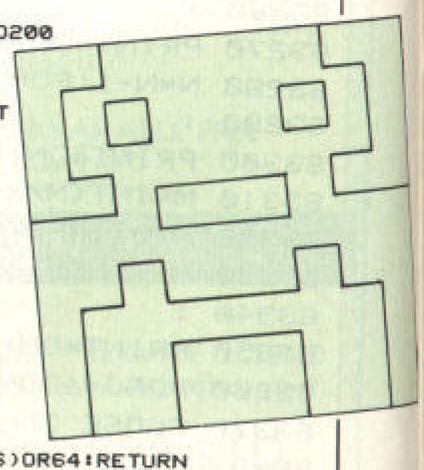
If you break out of the program, or create an error the screen may collapse — you will know what it is when it happens. If you want to get back to normal press stop/reset and X as normal.

All other instructions are contained in the program.

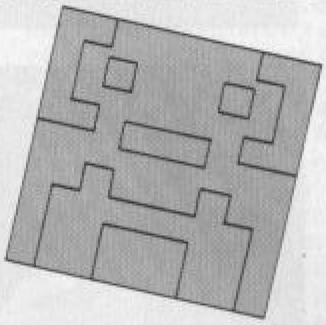
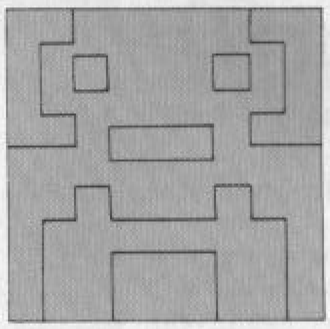
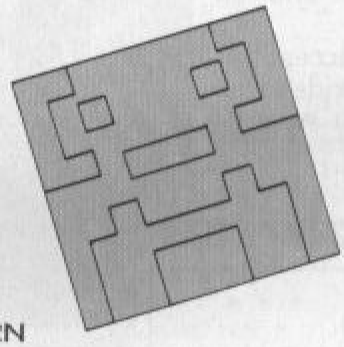
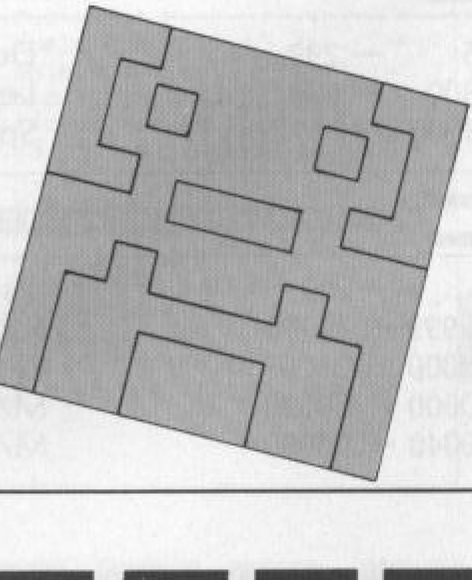
Program Listing



```
10 TRAP 920
20 GOSUB490
30 COLOR1,2,5:COLOR4,2,0:COLOR0,2,0:GOSUB600
40 DO:GETA$:LOOP UNTIL A$=""
50 PRINT"INPUT EVILNESS (1-9,1=VILE,9=EASY)":
60 PRINT"PRESS ANY OTHER KEY FOR REALLY VILE";
70 GETKEYL$:IFVAL(L$)>9THEN70
80 IFVAL(L$)=0THENFL=1
90 SC=3072:GOSUB870
100 GOSUB350:FORI=SC+960TOSC+1000:POKEI,32:NEXT:H=VAL(L$)+1:Y=SC+920:RV=128
110 Y1=29+RV:Y2=30+RV:Y3=31+RV:T=SC+81:R1=SC+957:L1=Y:R=23:O=12:F=55:C=32:P=198
120 SC=3072:AL=27+RV:LZ=28+RV
130 A1=INT(RND(TI)*38+T):A2=INT(RND(TI)*38+T):B1=INT(RND(TI)*3+39)
140 B2=INT(RND(TI)*3+39):POKEY,Y1:POKEY+1,Y2:POKEY+2,Y3
150 L=PEEK(P):IFL<>FANDL<>RANDL<>OTHEN190
160 IFL=FTHEN280
170 IFL=RANDY<R1THENPOKEY,C:Y=Y+1:POKEY,Y1:POKEY+1,Y2:POKEY+2,Y3:GOTO190
180 IFL=OANDY>L1THENPOKEY+2,C:Y=Y-1:POKEY,Y1:POKEY+1,Y2:POKEY+2,Y3
190 G=G+1:IFG>HTHENG=1
200 ONGGOTO210,220,150,150,150,150,150,150,150,150,150
210 A=A1:B=B1:GOSUB230:A1=A:GOTO150
220 A=A2:B=B2:GOSUB230:A2=A:GOTO150
230 A=A+B:SP=PEEK(A):IFSP=CORSP=ALTHENPOKEA-B,C:POKEA,AL:RETURN
240 GOSUB490
250 PRINT"THE ALIENS HAVE LANDED."
260 PRINT"YOU ZAPPED";S;"GREENIES ON EVILNESS";H-1
270 PRINT"     PRESS RETURN FOR ANOTHER GAME":DO:LOOP UNTIL PEEK(198)=1:RU
N
280 M=Y-39:FORI=MTOTSTEP-40:IFPEEK(I)=CTHENPOKEI,LZ:NEXT:GOTO320
290 S=S+1:POKEI,42+RV:IFI=A1THENA1=INT(RND(TI)*38+T):B1=INT(RND(TI)*3+39):GOTO31
5
310 A2=INT(RND(TI)*38+T):B2=INT(RND(TI)*3+39)
315 IFS=25THEN970
320 J=I:FORI=MTOJSTEP-40:POKEI,C:NEXT:G=G+1:IFG>2THENG=1
330 IFFL=1THENB1=INT(RND(TI)*3+39):B2=INT(RND(TI)*3+39):GOTO200
340 GOTO200
350 DIMC%(2)
355 RESTORE470:PRINT"";:COLOR4,1,3:FORI=0TO2:READC%(I):NEXT
360 FORSH=0TO7
370 FORLV=0TO2:CL=C%(LV):COLOR1,CL,SH
380 PRINT"                                        ";
390 NEXTLV:NEXTSH
400 COLOR1,1,5:COLOR0,6,5:RETURN
410 DATA60,90,90,42,126,219,129,153:REM ALIEN
420 DATA8,8,8,8,8,8,8,8:REM LASER
430 DATA0,0,0,3,31,63,127,255:REM BASE LEFT
440 DATA102,231,255,255,255,255,255,255:REM BASE CENTRE
450 DATA0,0,0,192,248,252,254,255:REM BASE RIGHT
460 DATA255,255,255,255,255,255,255,255:REM BLANK
470 DATA7,15,14:COLO(U)RS
480 POKE65298,192:POKE65299,60:RETURN:  POKE65286,PEEK(65286)OR64:RETURN
490 POKE65298,196:POKE65299,208:POKE65286,27:RETURN
```



Program Listing (cont.)

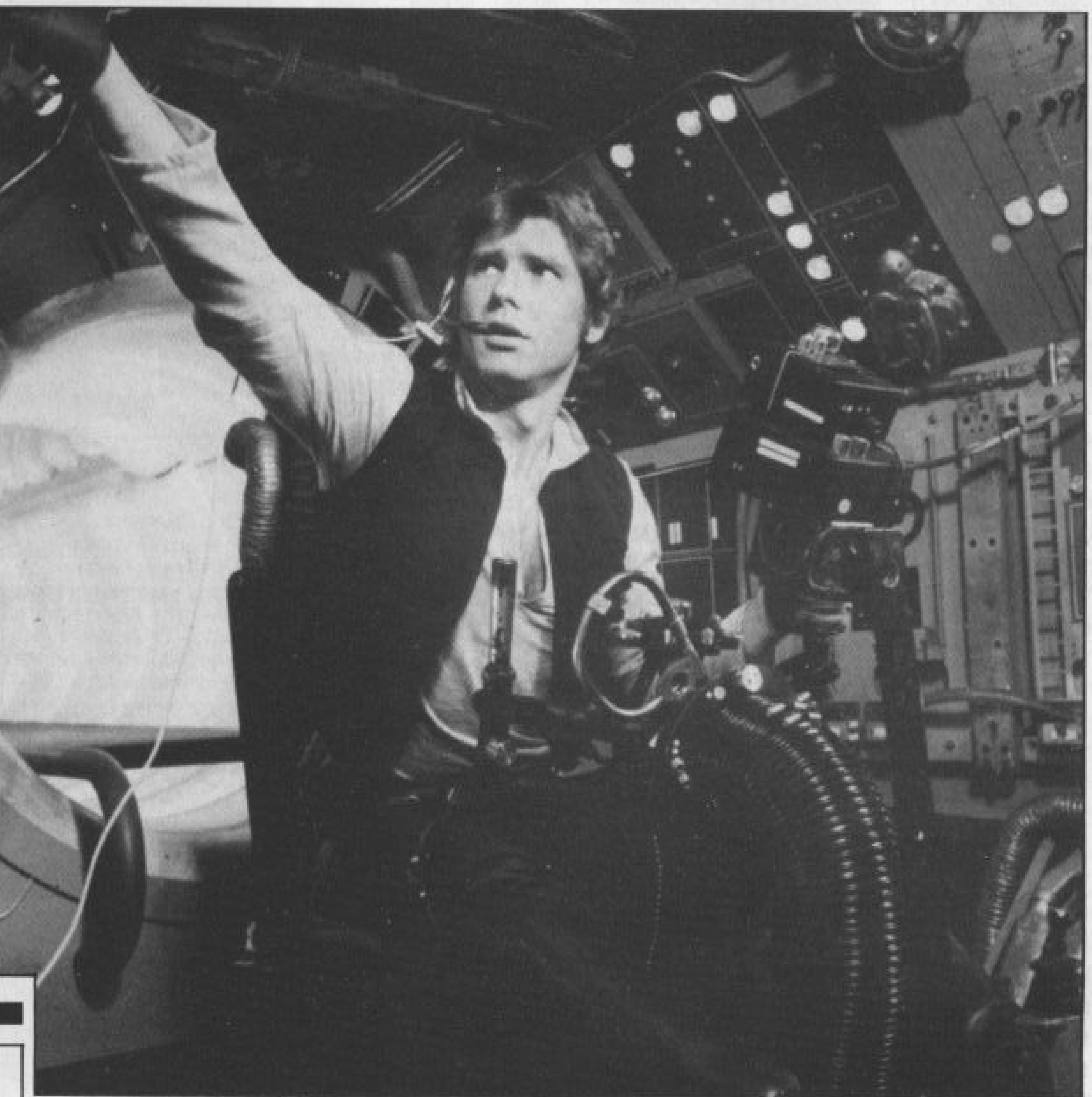
```
500 COLOR0,5,6:COLOR4,5,6:PRINT"         PRESS RETURN THREE TIMES"
510 FORI=0TO2
520 DO:LOOP UNTIL PEEK(198)=1
530 DO:LOOP UNTIL PEEK(198)<>1
540 PRINTI+1
550 NEXT:COLOR1,2,2:COLOR0,2,2:COLOR4,2,2
560 PRINT"T D000 D200 3C00"
570 PRINT"X":PRINT"G 590";
580 MONITOR
590 COLOR1,2,5:RETURN
600 PRINT"*THE VILE GREEN THINGS FROM OUTER SPACE";:COLOR1,2,5:PRINT"*"
610 PRINT" IN THIS GAME OF SKILL AND STRATEGY THE"
620 PRINT"PLAYER MUST BLAST TO BITS ALL OF THE"
630 PRINT"VILE GREEN ALIENS THAT ARE OUT TO GET"
640 PRINT"DEAR OLD MOTHER EARTH."
650 PRINT" THIS GAME CONCEPT IS TOTALLY ORIGINAL"
660 PRINT"(WHO WOULD THINK OF A GAME WHERE YOU"
670 PRINT"HAVE TO SHOOT GREEN ALIENS?)"
680 PRINT"BUT IF ANY OTHER READER HAS"
690 PRINT"SEEN A GAME WHERE YOU HAVE TO BLAST"
700 PRINT"ALIENS OUT OF THE SKY PLEASE LET US KNOW";
710 PRINT:PRINT"       PRESS ";:COLOR1,3,5:PRINT"RETURN TO CONTINUE":COL
OR1,2,5
720 DO:LOOP UNTIL PEEK(198)=1
730 GETA$
740 PRINT"*THE VILE GREEN THINGS FROM OUTER SPACE";:COLOR1,2,5:PRINT"*"
750 PRINT" YOU ARE THE LASER BASE AT THE BOTTOM OF THE SCREEN."
760 PRINT"THE VILE GREEN ALIENS ARE THE ALIENS"
770 PRINT"THAT ARE A VILE GREEN COLOUR."
780 PRINT" TO MOVE YOUR LASER BASE (WHICH IS ALSO"
790 PRINT"SICKLY GREEN COLOUR) YOU USE THESE"
800 PRINT"KEYS:"
810 PRINT"Z = LEFT"
820 PRINT"X = RIGHT"
830 PRINT"? = AHNIALATE THE VILE GREEN ALIENS"
840 PRINT" IF THEY GET TO THE BOTTOM THEN IT'S"
850 PRINT"HARD CHEESE AND YOU DIE IN GREAT AGONY"
860 PRINT"KILL LOTS OF THEM AND YOU GET POINTS":RETURN
870 GOSUB500:FORI=0TO47:READX:POKE15360+(27*8)+I,X:NEXT
880 GOSUB480:RETURN
890 GOSUB500
900 GOSUB480:PRINT"FRED":FORI=0TO63:POKE65299,I*4
910 FORJ=0TO500:NEXT:NEXT
920 GOSUB490:PRINT ERR$(ER)EL
930 COLOR0,2,7:COLOR4,7,5:COLOR1,1:END
970 GOSUB490:COLOR1,3,5:COLOR0,2,2
980 PRINT"YOU MADE IT!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
990 PRINT"YET AGAIN THE EARTH IS SAFE FROM THE"
1000 PRINT"ALIEN MENACE."
1010 PRINT"      PRESS ANY KEY FOR ANOTHER GAME"
1020 DO:FORI=0TO7:COLOR4,7,I:NEXT
1030 GETA$:LOOPUNTILA$=""
1035 GETKEYA$
1040 RUN

READY.
```

Have you got the nerve and courage to attack the aliens in this great arcade game from F G Tout. The race for space is on.

SCRAMBLE





Your Mission: To invade the Scramble system.
The Enemy: There are four waves of alien forces trying to prevent you from completing your mission. Each alien destroyed scores 100 points
WARNING: Try not to waste missiles or you will not have the ammunition to fire when you want to.
Your Reward: Should you complete the mission you will receive 2000 points and an extra space ship.
You do not have to accept this mission but if you decide to, all the information you need is listed below.
Good luck!

Variables

S1 — S9 = Sound
V Sprite Var (53248)
LJ/S :M1, A1, Q1 M/c Addresses
En Enemies : Sc score : Le level
R,S Joystick

Program Information Part 1

5 — 299	Download U.D.Gs
300 — 430	Letters
1000 —	Sprite data

Program Information Part 2

4 — 200	Instructions
19999 — 20080	M/C Joystick routine
30000 — 35009	Yet more sprite data
50000 — 50030	M/C scroll screen
50040 — 60000	M/C REMmed........

Program Information Part 3

5 — 99	Var's
1000 — 1999	Screen 1
2000 — 2170	Screen 2
3000 — 3170	Screen 3
4000 — 4170	Screen 4
5000 — 5035	Main Routine
5100 — 5270	Check Collisions
11000 — 12010	Hit Sound
13000 — 13020	Exit to next wave
20000 — 20010	Sprite positions
30000 — 30020	Game Over Routine
31000 — 31035	The End?????-------
40000 — 40115	Check Score
60000 — 60150	Title Page------
62000 — 62199	Musical Intro?
63000 — 63060	Key Ref.

Program Listing Part 1

```
9 REM CLR
10 PRINT"♥":POKE53280,2:POKE53281,2
20 POKE56334,PEEK(56334)AND254
30 POKE1,PEEK(1)AND251
40 FORG=0TO64*8:POKE12288+G,PEEK(53248+G):NEXT
50 POKE1,PEEK(1)OR4
60 POKE56334,PEEK(56334)OR1
100 READA%:T=T+1:IFA%<>-1THENPOKE12288+G+T,A%:GOTO100
101 FORI=0TO60*8STEP8:A=12288+1+I:B=PEEK(A):POKEA,B AND6:NEXT
102 DATA0,0,0,0,0,0
105 DATA255,126,60,60,60,126,255,204
110 DATA1,3,15,23,15,63,255,255
115 DATA128,208,244,248,250,252,254,255
120 DATA255,255,255,255,255,255,255,255
125 DATA0,0,0,16,56,16,0,0
130 DATA255,255,63,15,23,15,3,1
135 DATA255,254,252,250,248,244,209,128
140 DATA239,0,219,0,247,0,223,0,255,255,129,129,129,129,255,255
299 DATA-1
300 FORI=0TO214:READA:POKE12288+I,A:NEXT
305 DATA0,0,0,0,0,0,0,0
310 DATA48,124,108,126,118,102,102,0,48,124,108,126,118,102,126,0
320 DATA60,108,108,96,96,118,62,0,120,108,118,118,118,108,120,0
330 DATA126,118,96,124,96,118,126,0,126,118,96,124,100,112,112,0
340 DATA60,118,96,110,118,126,60,0,108,110,118,126,102,118,118,0
350 DATA124,56,24,28,28,30,62,0,62,30,28,12,108,120,56,0
360 DATA110,108,124,112,124,108,110,0,96,96,112,112,112,124,62,0
370 DATA54,119,127,107,119,119,119,0,54,118,122,126,111,119,119,0
380 DATA62,103,115,119,103,115,62,0,126,103,115,126,96,112,112,0
390 DATA62,103,115,115,115,62,15,0,126,115,103,126,124,102,103,0
400 DATA62,115,97,62,67,103,62,0,127,60,24,24,28,28,28,0
410 DATA103,103,71,103,102,126,60,0,103,103,103,71,102,62,28,0
420 DATA119,119,107,107,127,119,54,0,103,119,62,28,62,119,103,0
430 DATA115,51,62,24,24,28,28,0,126,15,12,24,48,127,63,0
1000 FORI=0TO255:READA:POKE213*64+I,A:NEXT
1010 DATA0,0,0,0,0,0,0,0
1011 DATA0,0,0,0,0,0
1012 DATA0,0,0,0,0,7,128
1013 DATA0,127,192,1,159,248,3,237,240,7,255,188,7,255
1014 DATA254,7,255,120,7,255,240
1015 DATA3,253,192,1,255,224,0
1016 DATA127,128,0,30,0,0,0
1017 DATA0,0,0,0,0,0,0
1018 DATA0,0,0,0,0,0
1019 DATA0,0,0,0,0,0
1020 DATA0,0,0,0,0,7,224
1021 DATA0,126,192,1,255,240,3
1022 DATA255,224,7,63,255,7,190
1023 DATA254,7,223,248,7,255,176
1024 DATA3,255,240,1,255,224,0
1025 DATA123,0,0,31,128,0,0
1026 DATA0,0,0,0,0,0,0
1027 DATA0,0,0,0,0,0
1028 DATA0,0,0,0,0,0
1029 DATA0,0,0,0,0,6,0
1030 DATA0,127,128,1,255,224,3,255,112,7,251,252,7,255
1031 DATA252,7,55,240,7,190,248
1032 DATA3,255,224,1,251,128,0
1033 DATA127,0,0,28,0,0,0
1034 DATA0,0,0,0,0,0,0
1035 DATA0,0,0,0,0,0
1036 DATA0,0,0,0,0,0
1037 DATA0,0,0,0,0,7,128
1038 DATA0,127,246,1,251,195,3
1039 DATA255,252,7,255,248,7,238
1040 DATA254,7,255,255,7,255,208
1041 DATA3,63,250,1,191,227,0
1042 DATA119,241,0,31,0,0,0
1043 DATA0,0,0,0,0,0,0
2000 FORI=0TO639:READA:POKE216*64+I,A:NEXT:REM**SAUCER***
2010 DATA0,0,0
2020 DATA0,0,0,0,0,0
2030 DATA0,0,0,0,0,0
2040 DATA0,0,0,0,0,0
2050 DATA0,0,0,0,0,0
2060 DATA0,0,60,0,0,126,0
2070 DATA0,255,0,1,255,128,15
2071 DATA255,240,57,153,156,249,153
2072 DATA159,63,255,252,15,255,240,0
2120 DATA0,0,0,0,0,0,0
2130 DATA0,0,0,0,0,0
2140 DATA0,0,0,0,0,0
2150 DATA0,0,0,0,0,0
2160 DATA0,0,60,0,0,126,0
2170 DATA0,255,0,1,255,128,15
2171 DATA255,240,59,51,60,243,51
2172 DATA63,63,255,252,15,255,240,0
2173 DATA0,0,0,0,0,0
2240 DATA0,0,0,0,0,0,0
2245 DATA0,0,0,0,0,0
2250 DATA0,0,0,0,0,0
2260 DATA0,0,60,0,0,126,0
2270 DATA0,255,0,1,255,128,15
2271 DATA255,240,62,102,124,246,102
2272 DATA111,63,255,252,15,255,240
2273 DATA0,0,0,0,0,0
2274 DATA0,0,0,0,0,0
2340 DATA0,0,0,0,0,0,0,0
2350 DATA0,0,0,0,0,0
2360 DATA0,0,60,0,0,126,0
2370 DATA0,255,0,1,255,128,15
2371 DATA255,240,57,153,156,249,153
2372 DATA159,63,255,252,15,255,240
2374 DATA0,0,0,0,0,0
2375 DATA0,0,0,0,0,0
2376 DATA0,0,0,0,0,0
2450 DATA0,0,0,0,0,0,0,0
2460 DATA0,0,60,0,0,126,0
2470 DATA0,255,0,1,255,128,15
2471 DATA255,240,59,51,60,243,51
2472 DATA63,63,255,252,15,255,240
2473 DATA0,0,0,0,0,0
2474 DATA0,0,0,0,0,0
2475 DATA0,0,0,0,0,0
2476 DATA0,0,0,0,0
2560 DATA0,0,60,0,0,126,0
2570 DATA0,255,0,1,255,128,15
2571 DATA255,240,57,153,156,249,153
2572 DATA159,63,255,252,15,255,240,0
2574 DATA0,0,0,0,0,0
2575 DATA0,0,0,0,0,0
2576 DATA0,0,0,0,0,0
2577 DATA0,0,0,0,0,0
2578 DATA0,0,0,0,0,0
2650 DATA0,0,0,0,0,0,0,0
2660 DATA0,0,60,0,0,126,0
2670 DATA0,255,0,1,255,128,15
2671 DATA255,240,59,51,60,243,51
2672 DATA63,63,255,252,15,255,240
2673 DATA0,0,0,0,0,0
2674 DATA0,0,0,0,0,0
2675 DATA0,0,0,0,0,0
2676 DATA0,0,0,0,0
2740 DATA0,0,0,0,0,0,0
2750 DATA0,0,0,0,0,0
2760 DATA0,0,60,0,0,126,0
2770 DATA0,255,0,1,255,128,15
2771 DATA255,240,57,153,156,249,153
2772 DATA159,63,255,252,15,255,240
2774 DATA0,0,0,0,0,0
2775 DATA0,0,0,0,0,0
2776 DATA0,0,0,0,0,0,0
2840 DATA0,0,0,0,0,0,0,0
2845 DATA0,0,0,0,0,0
2850 DATA0,0,0,0,0,0
2860 DATA0,0,60,0,0,126,0
2870 DATA0,255,0,1,255,128,15
2871 DATA255,240,62,102,124,246,102
```



Program 1 Listing (cont.)

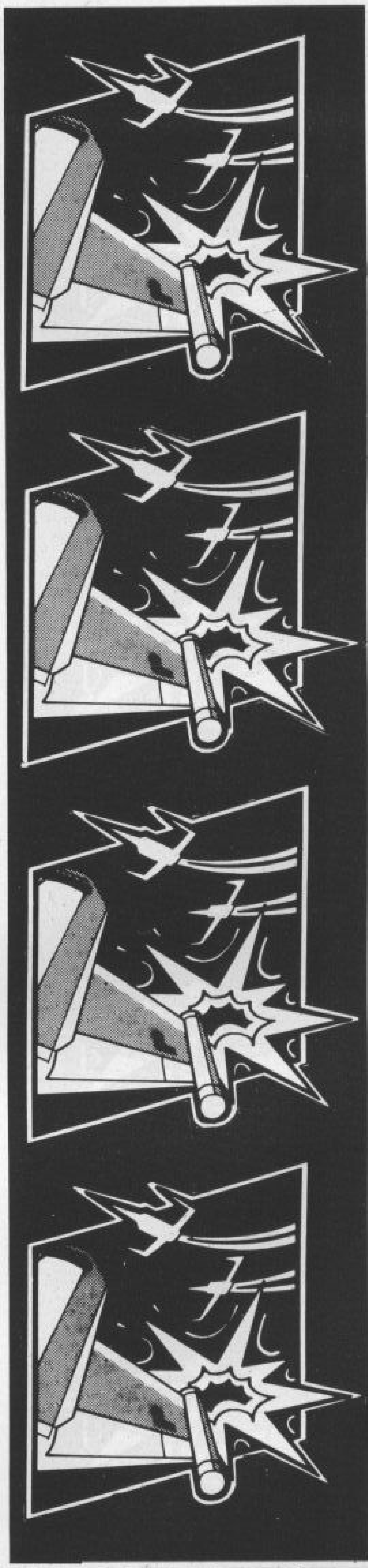
```
2872 DATA111,63,255,252,15,255,240
2873 DATA0,0,0,0,0,0
2874 DATA0,0,0,0,0,0,0
2920 DATA0,0,0,0,0,0,0,0
2930 DATA0,0,0,0,0,0,0
2940 DATA0,0,0,0,0,0,0
2950 DATA0,0,0,0,0,0,0
2960 DATA0,0,60,0,0,126,0
2970 DATA0,255,0,1,255,128,15
2971 DATA255,240,59,51,60,243,51
2972 DATA63,63,255,252,15,255,240,0
2973 DATA0,0,0,0,0,0,0
30000 FORI=0TO255:READA:POKE226*64+I,A:NEXT:REM**ROCKET
30010 DATA8,0,0,28,0,0,46,0,0,46
30020 DATA0,0,46,0,0,46,0,0,46,0,0,46,0,0
30030 DATA46,0,0,46,0,0,46,0,0,46,0,0,46,0
30040 DATA0,111,0,0,111,0,0,223,128,0,191,128,1,235,192,1,201,192,1,148,192,0
30050 DATA0,0,0,0,8,0,0,28,0,0,54,0,0,54
30060 DATA0,0,54,0,0,54,0,0,54,0,0,54,0,0
30070 DATA54,0,0,54,0,0,54,0,0,54,0,0,54,0
30080 DATA0,119,0,0,119,0,0,239,128,0,223,128,1,235,192
30090 DATA1,201,192,1,148,192,0
32051 DATA0,0,0,0,8,0,0
32052 DATA28,0,0,58,0,0,58
32053 DATA0,0,58,0,0,58,0
32054 DATA0,58,0,0,58,0,0
32055 DATA58,0,0,58,0,0,58
32056 DATA0,0,58,0,0,58,0
32057 DATA0,123,0;0,123,0,0
32058 DATA251,128,0,253,128,1,235
32059 DATA192,1,201,192,1,148,192,0
32061 DATA0,0,0,0,8,0,0
32062 DATA28,0,0,60,0,0,60
32063 DATA0,0,60,0,0,60,0
32064 DATA0,60,0,0,60,0,0
32065 DATA60,0,0,60,0,0,60
32066 DATA0,0,60,0,0,60,0
32067 DATA0,126,0,0,127,0,0
32068 DATA255,128,0,255,128,1,235
32069 DATA192,1,201,192,1,148,192,0
60000 POKE53280,5:POKE53281,5:POKE198,2:POKE631,13:LOAD
READY.
```

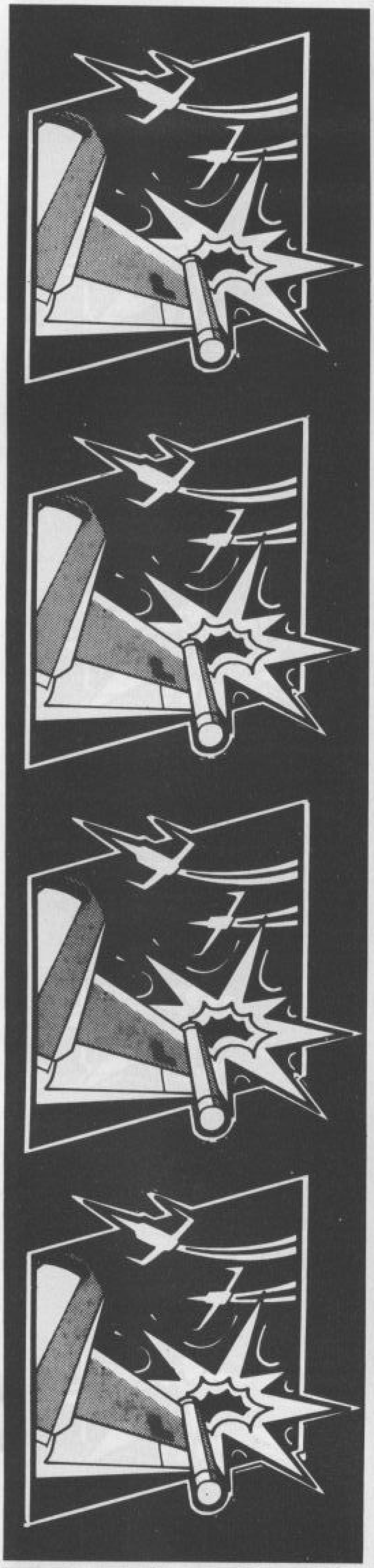
Program 2 Listing

```
3 REM CLR
4 PRINT"♥":POKE53281,4:POKE53280,4
5 GOTO19999
9 REM CLR
10 POKE53280,0:POKE53281,0:PRINT"♥";:POKE53272,29
19 REM CRD RED
20 PRINT"■        ■      SCRAMBLE BY F G TOUT.          ";
29 REM BLU
30 PRINT"         ■ ----------------------        ";
31 REM CRD-CYN
32 PRINT"▲ YOU CONTROL A POWERFUL SPACE SHIP,SENT ";
33 REM CRD-WHT RED WHT
34 PRINT"■■ TO INVADE THE ■SCRAMBLE■ SYSTEM.        ";
35 REM CRD-CYN
36 PRINT"▲ GUIDE YOUR SHIP THROUGH 4 WAVES OF      ";
37 REM CRD-WHT
38 PRINT"■■ ALIENS,AND DESTROY THEM.               ";
39 REM CRD-CYN
40 PRINT"▲ DO NOT COLLIDE WITH ANY ALIENS.....     ";
41 REM CRD-WHT
42 PRINT"■■ OR TERRAIN....AND SCORE 100 PTS FOR    ";
43 REM CRD-CYN
44 PRINT"▲ EACH ALIEN YOU DESTROY.                 ";
45 REM CRD-WHT
46 PRINT"■■ IF YOU MANAGE TO REACH THE FINAL WAVE..";
47 REM CRD-CYN
48 PRINT"▲ (AND GET THROUGH IT)....YOU WILL GET AS";
49 REM CRD-WHT
50 PRINT"■■ YOUR REWARD 2000 PTS + AN EXTRA SHIP.  ";
51 REM CRD-PUR
52 PRINT"■■       THERE'S MORE......PRESS SPACE     "
54 POKE198,0:WAIT197,60
59 REM CLR
60 PRINT"♥";
61 REM   2*CRD RED
62 PRINT"■■        ■      SCRAMBLE BY F G TOUT.         ";
63 REM BLU
64 PRINT"         ■ ----------------------        ";
```

Program 2 Listing (cont.)

```
65 REM CRD-CYN
66 PRINT"■ PLUG YOUR JOYSTICK INTO PORT TWO....     ";
67 REM CRD-WHT
68 PRINT"■■ THE CONTROLS ARE:-                       ";
69 REM CRD-CYN
70 PRINT"■ FOWARD .... UP : BACK .... DOWN           ";
71 REM CRD-WHT
72 PRINT"■■ LEFT ....BACK  : RIGHT ... FOWARD        ";
73 REM CRD-CYN
74 PRINT"■ FIRE/BUTTON TO FIRE MISSILE......         ";
75 REM CRD-WHT RED WHT
76 PRINT"■■ GOOD LUCK.......YOU ■WILL■ NEED IT.....   ";
77 REM CRD-CYN
78 PRINT"■ NOW PLEASE PRESS SHIFT/RUN-STOP TO LOAD"
79 REM CRD RED YEL BLU GRN
80 PRINT"■  ■SCRAMBLE ■SCRAMBLE ■SCRAMBLE ■SCRAMBLE  "
89 REM BLK
90 PRINT"■":END
200 END
19999 FORI=0TO68:READA:POKE49152+I,A:NEXT:REM**J/S**
20000 DATA 173,0,220,201,126,208,3,32,180,195,201,125,208,3,32,230
20010 DATA 195,201,123,208,3,32,24,196,201,119,208,3,32,74,196,201
20020 DATA 110,208,3,32,180,195,201,109,208,3,32,230,195,201,107,208
20030 DATA 3,32,24,196,201,103,208,3,32,74,196,32,80,195,32,92
20040 DATA 193,32.192,193,96
20050 FORI=0TO36:READA:POKE49300+I,A:NEXT:REM**MISSILE**
20060 DATA 173,2,208,201,0,240,30,169,255,141,21,208,238,2,208,238
20070 DATA 2,208,238,2,208,238,2,208,238,2,208,238,2,208,238,2
20080 DATA 208,238,2,208,96
30000 FORI=0TO191:READA:POKE210*64+I,A:NEXT
30001 DATA0,0,0,0,0,0,31
30002 DATA0,0,63,128,0,113,192
30003 DATA0,127,224,0,241,240,0
30004 DATA255,190,0,241,183,0,127
30005 DATA179,192,113,255,254,63,255
30006 DATA255,31,31,128,0,15,254
30007 DATA0,0,0,0,0,0,0
30008 DATA0,0,0,0,0,0,0
30009 DATA0,0,0,0,0,0,0,0
30011 DATA0,0,0,0,0,0,31
30012 DATA0,0,49,128,0,127,192
30013 DATA0,113,224,0,127,240,0
30014 DATA113,254,0,127,247,0,113
30015 DATA243,192,127,255,254,49,255
30016 DATA255,31,31,128,0,15,254
30017 DATA0,0,0,0,0,0,0
30018 DATA0,0,0,0,0,0,0
30019 DATA0,0,0,0,0,0,0,0
30020 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
30021 DATA0,0,0,47,0,1,91,252,0,181,255,1,75,252,0,47,0,0,0,0,0
30022 DATA0,0,0,0,0,0,0,0,0,0,0,0,0
35000 FORI=0TO63:READA:POKE235*64+I,A:NEXT
35001 DATA0,7,0,33,8,1,99
35002 DATA16,2,20,32,10,32,64
35003 DATA220,32,142,24,0,97,150
35004 DATA3,131,176,59,119,20,64
35005 DATA2■0,116,171,253,50,70,248
35006 DATA240,187,54,28,143,2,254
35007 DATA65,240,60,36,111,30,22
35008 DATA3,226,19,8,124,1,142
35009 DATA23,0,195,1,0,0,240,0
50000 FORI=0TO46:READA:POKE50000+I,A:NEXT
50010 DATA169,40,162,24,133,87,169,4,133,88,160,0,177,87,133,89,200,177,87,136
50020 DATA145,87,200,152,201,39,208,244,165,89,145,87,165,87,24,105,40,133
50030 DATA87,144,2,230,88,202,208,220,96
50040 FORI=0TO50:READA:POKE49500+I,A:NEXT:REM***MOVE & CHANGE ALIEN*****
50050 DATA162,4,189,0,208,24,233,2,157,0,208,201,245,144,6,173,4,220,157
50060 DATA1,208,232,232,224,16,208,231
50070 DATA162,2,189,248,7,24,105,1,201
50080 DATA212,144,2,169,210,157,248,7
50090 DATA232,224,8,208,236,96,0
50100 FORI=0TO23:READA:POKE49600+I,A:NEXT:REM***CHANGE ME*****
50170 DATA162,0,189,248,7,24,105,1,201
50180 DATA212,144,2,169,210,157,248,7
50190 DATA232,224,1,208,236,96,0
52040 FORI=0TO50:READA:POKE49700+I,A:NEXT:REM***MOVE & CHANGE ALIEN†*****
52050 DATA162,4,189,1,208,24,233,2,157,1,208,201,245,144,6,173,4,220,157
52060 DATA0,208,232,232,224,16,208,231
52070 DATA162,2,189,248,7,24,105,1,201
52080 DATA212,144,2,169,210,157,248,7
52090 DATA232,224,8,208,236,96,0
53000 FORI=0TO19:READA:POKE50100+I,A:NEXT:REM***ME  ↑**
53010 DATA173,1,208,201,60,240,12,206,1,208,206,1,208,206,1,208,206,1,208,96
53020 FORI=0TO19:READA:POKE50150+I,A:NEXT:REM***ME DOWN***
53030 DATA173,1,208,201,200,240,12,238,1,208,238,1,208,238,1,208,238,1,208,96
53040 FORI=0TO31:READA:POKE50200+I,A:NEXT:REM***ME ←***
```





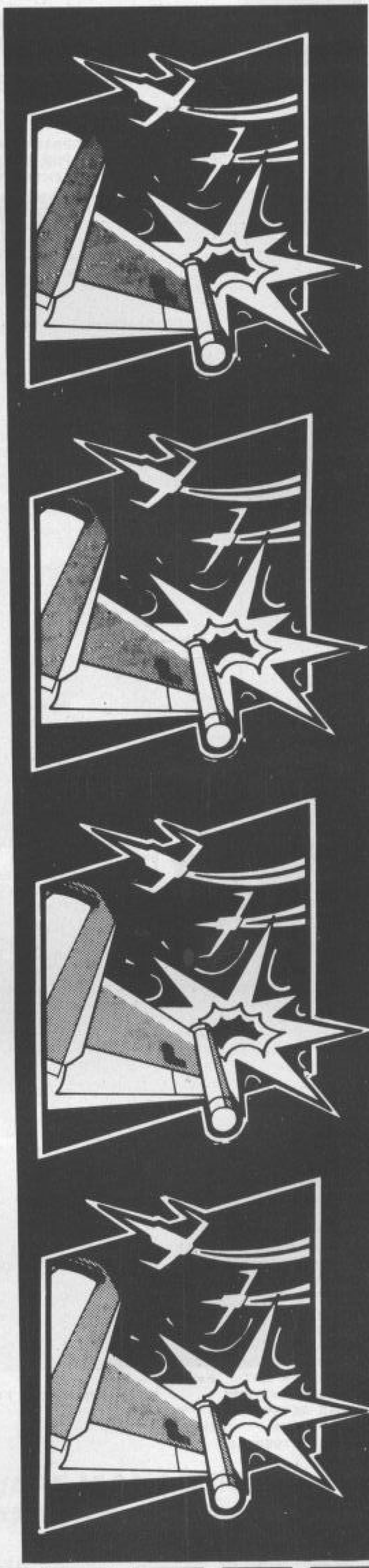
```
53050 DATA173,0,208,201,28,240,24,206,0,208,206,0,208,206,0,208
53060 DATA206,0,208,206,0,208,206,0,208,206,0,208,206,0,208,96
53070 FORI=0TO31:READA:POKE50250+I,A:NEXT:REM***ME RIGHT***
53080 DATA173,0,208,201,172,240,24,238,0,208,238,0,208,238,0,208
53090 DATA238,0,208,238,0,208,238,0,208,238,0,208,238,0,208,96
60000 GOTO10
READY.
```

Program 3 Listing

```
5 S1=54276:S2=54277:S3=54273:S4=S1+7:S5=S2+7:S6=S3+7:S7=S4+7:S8=S5+7:S9=S6+7
6 FORI=54272TO54296:POKEI,0:NEXT:POKE54296,15
10 V=53248:POKE53272,29:L=56320:M1=49300:A1=49152:SC=0:HI=0:LI=3:LE=1
20 POKEV+21,253:POKE2040,210:POKE2041,212
30 GOSUB60000:EN=0:SC=0:LE=1
90 Q1=49152:XX=20:DEFFNB(B)=(PEEK(V+30)AND1):DEFFNA(A)=(PEEK(V+31)AND1)
95 R=100:S=112:V1=53269:Z=253:DD=0
99 DD=0:LI=3
100 GOTO1000
1000 POKE53280,0:POKE53281,9:POKE49215,36:POKE49216,194:EN=0:SP=227
1002 IFDD=2THEN30000
1004 REM CLR- 13*CRD
1005 PRINT"■■■■■■■■■■■■■■";
1009 REM BWN
1010 PRINT"■            |—                              ";
1020 PRINT"           |——                               ";
1030 PRINT"          |——————|—                          ";
1040 PRINT"         |—————————   |—                     ";
1045 PRINT"        |———————————|——        |—            ";
1050 PRINT"       |——————————————        |——   |—       ";
1055 PRINT"      |————————————————|—   |————|——         ";
1060 PRINT"|——————————————————————|——————————|—         ";
1065 PRINT"————————————————————————————————————————|";
1069 REM BLK
1070 PRINT"————————————————————————————————————————■"
1080 FORI=1944TO2023:POKEI,68:POKEI+54272,9:NEXT:POKEV+27,252:AQ=49700:POKE53281
,0
1090 POKEV+39,6:POKE49736,230:POKE49740,226:POKE49707,6:POKEV+1,144:POKEV,28
1095 GOSUB20000:POKEV+30,0:POKEV+31,0:POKE49536,230:POKE49540,226:GOSUB40000
1902 GOSUB5000:IFFNB(B)=1ANDPEEK(V+30)>4THENGOSUB11000:GOTO1000
1903 IFEN>ECTHENGOSUB13000:GOTO2000
1904 IFFNA(A)=1THENGOSUB11000:GOTO1000
1999 GOTO1902
1999 REM CLR-CRD-PUR
2000 POKE53280,0:POKE53281,0:PRINT"■■■";:POKE53272,29:POKEV+1,144:POKEV,28
2001 POKEV+39,6:EN=0:POKE49215,92:POKE49216,193:LE=2:SP=220
2002 IFDD=2THEN30000
2005 PRINT"———————————————————————————————————————";
2010 PRINT"———————————————————————————————————————";
2015 PRINT"————————————————————————————————|_———";
2020 PRINT"_————|_——————|_|_—————————|   _|_|";
2025 PRINT" _|_——|   _——————|     _——|_————|       ";
2030 PRINT"     _|     _————|_|     _|  _——|        ";
2035 PRINT"          _——|               _|          ";
2040 PRINT"           _|                            ";
2044 REM   6*CRD
2045 PRINT"■■■■■■";
2063 PRINT"        _ |—                             ";
2064 PRINT"—         |——                       |—|";
2065 PRINT"——       |—————      |—       |—  |————";
2070 PRINT"———  |——————       |———      |———|————";
2085 PRINT"————|————————     |————    |——|————————";
2087 PRINT"———————————————   |————————|———————————";
2090 PRINT"——————————————|————————————————————————";
2095 PRINT"———————————————————————————————————————";
2099 FORI=1944TO2023:POKEI,68:POKEI+54272,0:NEXT:POKEV+27,252:AQ=49700
2100 AQ=49500:POKE49507,8:POKE49536,226:POKE49540,216:GOSUB40000
2150 GOSUB20000:POKEV+5,80:POKEV+7,90
2162 GOSUB5000:IFFNB(B)=1ANDPEEK(V+30)>4THENGOSUB11000:GOTO2000
2163 IFEN>ECTHENGOSUB13000:GOTO3000
2165 IFFNA(A)=1THENGOSUB11000:GOTO2000
2170 GOTO2162
3000 V=53248:SP=214
3002 POKE53280,0:POKE53281,0:IFDD=2THEN30000
3005 POKE49536,216:POKE49540,213:POKE49507,12:FORI=41TO46:POKE53248+I,2:NEXT
3006 FORI=2042TO2047:POKEI,213:NEXT:POKE53269,253:GOSUB40000:POKEV+39,6:LE=3
3009 REM GRN
3010 PRINT"■———————————————————————————————————————";
3011 REM GRN
```

Program 3 Listing (cont.)

```
3012 PRINT"■──────────────────────────────────────";
3015 REM GRN
3016 PRINT"■──────────────────────────────────────";
3020 PRINT"────────────|─────────────────|──|─";
3025 PRINT"─────────|    ─|─────────────|   ─|   ─";
3030 PRINT" ─────────|          ──|─♥♥♥♥♥─♥|        ";
3035 PRINT"  ───────|          ─|   ─♥♥♥♥─|         ";
3040 PRINT"   ────|                  ─♥♥─|          ";
3045 PRINT"    ──|                  .──|            ";
3050 PRINT"     ─|                                  ";
3054 REM   3*CRD
3055 PRINT"■■■";
3060 PRINT"                    |─                   ";
3062 PRINT"                    |───   |─            ";
3065 PRINT"              |─    |─────|───           ";
3070 PRINT"             |───  |──────────     |─    ";
3080 PRINT" |─    |─  |─────|──────────   |───|─   ";
3085 PRINT"|───|─|───|─────────────────|──────";
3090 PRINT"───────────────────────────────────────";
3092 PRINT"───────────────────────────────────────";
3094 PRINT"───────────────────────────────────────";
3100 AQ=49500:V=53248:EN=0:POKEV,28:POKEV+1,144
3150 GOSUB20000:POKEV+5,80:POKEV+7,90:POKEV+30,0:POKEV+31,0
3162 GOSUB5000:IFFNB(B)=1ANDPEEK(V+30)>4THENGOSUB11000:GOTO3000
3163 IFEN>ECTHENGOSUB13000:GOTO4000
3165 IFFNA(A)=1THENGOSUB11000:GOTO3000
3170 GOTO3162
3999 REM CLR-RED-CRD
4000 POKE53280,11:POKE53281,0:POKE53272,29:PRINT"■■■";:LE=4:SP=227
4002 A$="|||||||||||||||||||||||||||||||||||||||":PRINTA$;:EN=0
4003 IFDD=2THEN30000
4010 POKE49707,8:FORI=2042TO2047:POKEI,227:NEXT:V=53248:POKEV+21,253
4012 FORI=41TO46:POKEV+I,6:NEXT:POKE49215,36:POKE49216,194
4025 PRINT"|||||||||||||||▚▚▚▚▚▚||||▚▚▚▚▚||▚▚▚▚|";
4030 PRINT"●●●●●          |||||||||||||||||||||||||| ";
4031 PRINT"               |||||||               || ";
4032 PRINT"               |||||||                  ";
4033 PRINT"               |||||||                  ";
4034 PRINT"|||            |||||          ●●●       ";
4036 PRINT"||||             ||           ||||       ";
4038 PRINT"▚▚||                         |||||       ";
4040 PRINT"|||||                      |||||||       ";
4042 PRINT"|||||                      ||▚▚||       ";
4045 PRINT"▚▚||                       |||||||       ";
4050 PRINT"||||●●●●●●●●●●●●●●●●●●●●●●▚▚▚▚▚▚||▚▚|||";
4055 PRINT"|||||||||||||||||||||||||||||||||||||||";
4065 PRINT"|||||||||||||||||||||||||||||||||||||||";
4075 PRINT"▚▚|||||▚▚▚▚|||▚▚|||▚▚▚▚▚▚▚▚|▚▚▚▚|||";
4076 PRINTA$;A$;
4085 PRINT"▚▚|||▚▚▚▚▚▚▚▚|||||▚▚▚▚▚▚▚▚|||||▚▚▚▚||";
4086 PRINTA$;A$;
4095 PRINT"|||||||||||||||||||||||||||||||||||||||";
4100 POKEV+39,6:POKEV,28:POKEV+1,80:GOSUB40000
4150 GOSUB20000:POKEV+30,0:POKEV+31,0
4162 GOSUB5000:IFFNB(B)=1ANDPEEK(V+30)>4THENGOSUB11000:GOTO4000
4163 IFEN>ECTHENGOSUB13000:GOTO31000
4165 IFFNA(A)=1THENGOSUB11000:GOTO4000
4170 GOTO4162
5000 SYS49152:IFPEEK(L)<112ANDPEEK(L)>100THEND=1
5020 IFD=1THEN5030
5025 POKEV1,Z:POKEV+2,PEEK(V):POKEV+3,PEEK(V+1):RETURN
5030 POKEV1,Z+2:IFPEEK(V+2)<235THENSYSM1:SYSM1:SYSM1:GOSUB5100:RETURN
5035 POKEV1,Z:D=0:RETURN
5100 JJ=PEEK(V+30):IF(JJAND2)=2ANDJJ>5ANDJJ<131THENGOSUB5200:RETURN
5150 RETURN
5200 POKES1,128:POKES2,44:POKES3,5:POKES1,129:SYS50000
5210 IFJJ=130THENF1=2047:F2=15:F3=14
5220 IFJJ=6THENF1=2042:F2=5:F3=4
5230 IFJJ=10THENF1=2043:F2=7:F3=6
5240 IFJJ=18THENF1=2044:F2=9:F3=8
5250 IFJJ=34THENF1=2045:F2=11:F3=10
5260 IFJJ=66THENF1=2046:F2=13:F3=12
5265 SC=SC+100:GOSUB40100:EN=EN+1
5270 POKEF1,235:D=0:POKEV+F2,80+INT(RND(.)*100):POKEV+F3,255:RETURN
11000 POKES1,0:POKES2,31:POKES4,0:POKES5,14:LI=LI-1:IFLI<1THENDD=2
11010 POKE53281,1:FORTT=30TO5STEP-1
12000 POKES3,TT:POKES6,TT:POKES1,33:POKES4,33:NEXTTT:FORT=2TO255STEP3:POKES3,T
12004 REM CLR
12005 POKES1,129:NEXTT:FORT=0TO1000:NEXT:POKE53281,2:POKE53280,2:PRINT"■"
12010 POKEV+21,0:POKES1,0:POKES2,44:POKES3,5:POKES1,129:FORT=0TO20:NEXT:RETURN
13000 POKEV+21,1:POKEV+29,1:CC=1:FORI=PEEK(V+1)TO60STEP-2:POKEV+1,I:POKEV+39,C
13002 POKEV+21,1:POKEV+29,1:CC=1:FORI=PEEK(V+1)TO60STEP-2:POKEV+1,I:POKEV+39,C
13005 C=C+1:POKES1,32:POKES2,31:POKES3,255-I:POKES1,33:SYS50000:NEXT
```



Program 3 Listing (cont.)

```
13010 POKES1,128:POKES2,15:POKES3,5:POKES1,129:POKES4,128:POKES5,44:POKES6,10:PO
KES4,129
13020 POKEV+39,7:FORT=PEEK(V)TO254STEP4:POKEV,T:SYS50000:NEXT:POKEV+29,0:RETURN
20000 GG=255:FORI=5TO15STEP2:GG=GG-3:POKEV+I,GG:POKEV+I-1,GG:NEXT
20010 FORI=2042TO2047:POKEI,SP:NEXT:RETURN
29999 REM CLR
30000 PRINT"■":POKEV+21,0:Q$="                    GAME  OVER              "
30005 POKES2,44:POKE53281,0
30006 REM HOM-  2*CRD
30007 FORTT=0TO14:PRINT"■■■";:POKES1,32
30010 FORT=0TO9:PRINTQ$;:POKE646,INT(RND(1)*15):POKES3,T+10:POKES1,33:NEXTT,TT
30020 FORT=0TO1000:NEXT:GOTO30
30999 REM CLR HOM-  9*CRD
31000 PRINT"■";:POKEV+21,0:PRINT"■■■■■■■■■■";:POKES1,128:POKES2,44:POKES3,5:POKE
S1,129
31009 REM CYN
31010 PRINT"■   CONGRATULATIONS , YOU HAVE INVADED   ";
31019 REM WHT-CRD RED WHT BLU WHT
31020 PRINT"■■ THE SYSTEM.......■ONE SHIP■ AND ■2000 ■PTS ";
31029 REM CYN-CRD
31030 PRINT"■■ AWARDED.......HERE WE GO AGAIN.......  ";
31035 FORT=0TO5000:NEXT:LI=LI+1:SC=SC+2000:GOTO1000
39999 REM HOM YEL BLU GRN ORN
40000 PRINT"■";:PRINTTAB(1)"■SC "TAB(12)"■LI"TAB(18)"■LE"TAB(24)"■HI-SC"
40100 IFSC>HITHENHI=SC
40109 REM HOM-CYN
40110 PRINT"■■";:PRINTTAB(5);SC;TAB(14);LI;TAB(20);LE;TAB(30);HI
40115 RETURN
59999 REM CLR-  8*CRD
60000 V=53248:POKE53280,9:POKE53281,14:PRINT"■■■■■■■■■";:POKEV+21,0:POKE53272,29
60009 REM BLU
60010 PRINT"■|—— |—— |—— |—— |-  |- |—— |   |——";:EC=15
60015 PRINT"-    -    - - - - ——|—— - - -   -    ";
60020 PRINT"-    -    - - - - -|-|- - - -   -    ";
60025 PRINT"_——  -    ——| ———- - ——| - -  ——     ";
60030 PRINT"   - -    - _ - - -   - - - -   -    ";
60035 PRINT"   - -    - - - - -   - - - -   -    ";
60040 PRINT"-——| —— | - | - | -   | ——| —— ——|    ";
60045 FORT=0TO79:SYS50000:NEXT:POKES1,128:POKES2,44:POKES3,10:POKES1,129
60050 POKE53280,0:POKE53281,0
60054 REM   2*CRD RED PUR GRN BLU YEL RED PUR
60055 PRINT"■■                  ■BY ■F ■G ■TO■D■U■T":POKE198,0
60056 REM   2*CRD BWN LGN PUR
60057 PRINT"■■                  ■F.7■ FOR ■GAME."
60057 REM CRD GRN PUR BLU YEL CYN RED BWN
60058 PRINT"■                   ■HIGH ■S■C■O■R■E■"HI
60058 REM HOM PUR WHT GRN YEL
60059 PRINT"■               ■K ■= ■KEY ■REFERNCE"
60059 REM HOM-  2*CRD GRN YEL RED PUR
60060 PRINT"■■■               ■ALIENS■ PER ■WAVE■"EC
60100 GETA$:IFA$=""THEN60100
60109 REM F7
60110 IFA$="■"THENGOSUB62000:RETURN
60114 REM F1
60115 IFA$="■"THENEC=EC+5:IFEC>95THENEC=20
60119 REM HOM
60120 IFA$="■"THENEC=15
60125 IFA$="K"THEN63000
60150 GOTO60060
62000 REM
62005 POKES2,15:POKES5,15:POKES8,15:POKES1,32:POKES4,32:POKES7,16
62010 READA,B,C,D
62040 POKES3,A:POKES6,B:POKES9,C:FORT=0TOD*15:NEXT
62050 POKES1,33:POKES4,33:POKES7,17
62055 IFA=0THENRESTORE:RETURN
62060 GOTO62010
62100 DATA10,10,5,20,10,10,5,20,15,15,10,20,20,20,30,30,20,25,15,60
62105 DATA10,10,5,20,10,10,5,20,15,15,10,20,20,20,30,30,20,25,15,100
62110 DATA10,10,20,20,5,5,5,100
62199 DATA0,0,0,0
62999 REM CLR-  6*CRD
63000 POKE53280,12:POKE53281,11:PRINT"■■■■■■■"
63009 REM CYN
63010 PRINT"■  F 1 INCREASE ALIENS PER WAVE"
63019 REM WHT-CRD
63020 PRINT"■■  F 7 START GAME"
63029 REM CYN-CRD
63030 PRINT"■■  HOME  RESET"
63039 REM WHT-  2*CRD
63040 PRINT"■■■         PRESS SPACE FOR TITLE PAGE"
63045 GETX$:IFX$=""THEN63045
63050 IFX$=" "THENGOTO60000
63060 GOTO63045
READY.
```



NOUGHTS & CROSSES

Test your prowess against the computer in B. Davis' computerised version of noughts and crosses for the Commodore 64 or VIC 20.

NOUGHTS AND CROSSES has appealed to people of all ages for many years. Now you can test your wits against your computer but, beware, computers never make mistakes!

This program offers three levels of play. On the lowest level (Novice), the computer makes a logical move 38% of the time. Level 2 (Amateur) makes the computer play logically 68% of the time. And, in order that you don't get too frustrated, the computer will play logically only 98% of the time on Level 3. I suggest that you play at this level on your friend's computer!

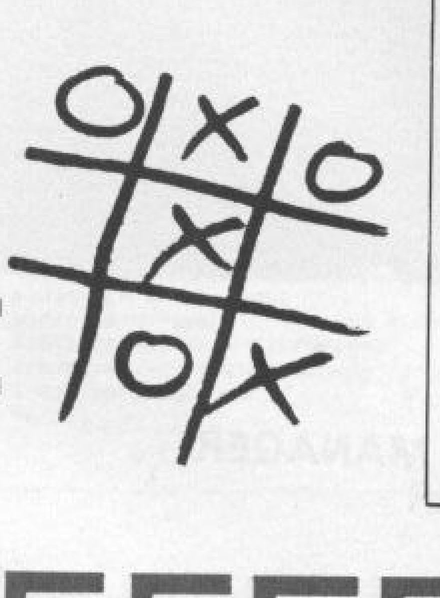
Type in the program exactly. It is designed to run on any VIC 20 (with or without memory expansion) or a CBM 64.

The game initially asks you to select which level you wish to play and whether you wish to play first. From then on, you and the computer will take it in turns to go first. The first player is always 'X' and, while the computer is making its move, its 'thoughts' are displayed. A score tally is displayed at the end of every game.

Line Explanation

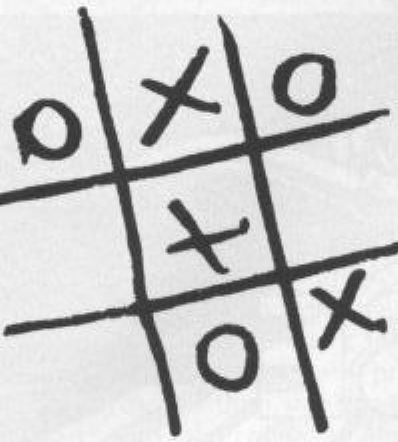
- 70-110 Find out where the screen is. CBM 64 = 1024, 7680 is a VIC (3.5K or 6.5K) and 4096 is a VIC 8/16K. Initialise screen, colour and sound variables.
- 120-190 Initialise scores, print title. Decide level of play and who starts game.
- 200-210 Start game. Set up 'X' and '0' POKE values. Print message.
- 230-270 Game loop. Call subroutines for sound, computer's move, win/lose/draw check, opponent's move. Exit if flag is not 0
- 280-360 End game. Print appropriate message. Call 'Flash' routine (if win). Print score table. Display 'Play Again?'. If 'NO', call END routine.
- 370-380 Decide who goes first. Go to START (200).
- 390-430 SUB: WIN/LOSE/DRAW scan. Set flag accordingly.
- 440-470 SUB: Flash winning line
  SUB: Computer's move
- 480-530 Randomise Chance (1-100) and move (1-9). If move is occupied, find another. If chance value is greater than level (38%, 68% or 98%), then play that move. Print 'thoughts'.
- 540-660 Follow through logical fields and decide which move is to be made. The secret here is in the DATA statements where all the possible combinations are each checked in turn, accepting the highest field result returned. Print 'thoughts'.
- 670 Play the move. Flash and sound.
- 680-760 SUB: Print grid
- 770 SUB: PEEK for current values in grid.
- 780-790 SUB: Sound routine
- 800 SUB: Delay loop
- 810-820 SUB: Clear sound registers. Used for CBM 64 only.
- 830-840 SUB: Flash message for wrong input from opponent.
- 850 SUB: Long delay loop
- 860 END routine.
- 870-890 DATA values





Program Listing

```
10 REM   NOUGHTS
20 REM          AND
30 REM             CROSSES
40 REM
50 REM   BY B.DAVIS SEPT'84
60 REM
70 SR=PEEK(648)*256:IFSR=1024THENCR=54272:POKE53281,14:GOTO100
79 REM CRD BLK
80 CR=30720+(3072ANDSR=4096):POKE36879,59:POKE36875,240:CA$="■ ":TB=0:CU$="■"
90 FORL=0TO2:S(1+L)=272+SR+L*2:S(4+L)=316+SR+L*2:S(7+L)=360+SR+L*2:NEXT:GOTO120
99 REM BLU
100 FORL=0TO2:S(1+L)=1522+L*2:S(4+L)=1602+L*2:S(7+L)=1682+L*2:NEXT:TB=10:CU$="■"
109 REM CRD
110 GOSUB810:POKECR,150:POKECR+1,69:POKECR+5,15:POKECR+24,15:CA$="■
119 REM CLR-WHT HOM- 19*CRD HOM-  2*CRD
120 PRINT"■■":SC=0:SO=0:SD=0:CD$="■■■■■■■■■■■■■■■■■■■":CC$="■■"+CA$
129 REM   2*CRD   2*CRD   2*CRD
130 PRINTCA$"NOUGHTS■■AND■■CROSSES":PRINTCA$"■■"CU$"WHAT LEVEL OF PLAY ?"
140 PRINTCA$"1. NOVICE":PRINTCA$"2. AMATEUR":PRINTCA$"3. PROFESSIONAL"
150 GETK$:IFK$=""THEN150
160 LV=VAL(K$):IFLV<1ORLV>3THEN150
169 REM WHT CRL
170 PRINT"■"LEFT$(CD$,10+LV*2)CA$"■"LV
179 REM   2*CRD
180 PRINTLEFT$(CD$,18)CA$"■■YOU GO FIRST ? (Y:N)"CU$
190 GETK$:IFK$<>"Y"ANDK$<>"N"THEN190
200 GOSUB680:IFK$="Y"THENO=24:C=15:MT=1:PRINTCA$"I AM  O ..YOU ARE  X":GOTO230
210 O=15:C=24:MT=0:PRINTCA$"I AM  X ..YOU ARE  O"
220 PRINTCC$"MY GO..  ":GOSUB480:GOSUB390:IFF>0THEN280
229 REM   2*CRL
```



How it works

No REMarks are to be included in the listing to keep the program small enough to fit in the 3.5K VIC.

The computer's 'thinking' is based on three fields, in this precise order:

1. If a winning line can be played, play it.
2. If a block is required, block it.
3. Attempt to occupy any three of the four corners.

Variables

SR	Screen RAM
CR	Colour RAM
TB	Tab value
SC	Score: Computer
SO	Score: Opponent
SD	Score: Draw
LV	Level of play
MT	Move tally
CH	Chance value
CM	Computer's move
F1	Flash Location
F2	Flash location
F3	Flash Location
CA$	Cursor Advance
CC$	Cursor Combination
CD$	Cursor Down
CU$	Colour
L	Loop
M	Outer Loop
O	Opponent's code
C	Computer's Code
D	Delay loop
K	Value of K$
W	Win value
R	Random Move
G	POKE value X/O
H	Temporary store
X	Data value
Y	Data value
Z	Data value
F	Flag
K$	Key input
S(9)	Grid locations
P(9,	Peek values of grid

Note: Although CR is the colour RAM, when run on a CBM 64, its value is conveniently 54272 which is also used for the sound register off-set, lines 110 and 810.

Program Listing

```
230 POKE649,1:GOSUB800:GOSUB780:PRINTCC$"YOUR GO..NUMBER ?   ■■";
240 GETK$:IFK$<"1"ORK$>"9"THEN240
249 REM WHT
250 POKE649,0:GOSUB780:PRINT"■"K$CU$:K=VAL(K$):IFP(K)<>48+KTHENGOSUB830:GOTO230
259 REM HOM-  2*CRD
260 POKES(K),0:POKES(K)+CR,1:PRINT"■■■"CA$"                    "
270 GOSUB390:POKE649,10:IFF=0THEN220
280 ONFGOTO290,300,310
290 PRINTCD$TAB(7+TB)"YOU WIN":GOSUB440:SO=SO+1:GOTO320
300 PRINTCD$TAB(8+TB)"I WIN":GOSUB440:SC=SC+1:GOTO320
310 PRINTCD$TAB(5+TB)"IT'S A DRAW":SD=SD+1
320 PRINTCA$"PLAY AGAIN ? (Y/N)"
329 REM HOM CRL   2*CRL
330 PRINT"■";CA$"  SCORE (LEVEL"LV"■)    ■■"
339 REM HOM-  2*CRD
340 PRINT"■■■"CA$"ME      YOU     DRAW":PRINTCA$SC"     "SO"     "SD
350 GETK$:IFK$<>"Y"ANDK$<>"N"THEN350
359 REM WHT
360 IFK$="N"THENK$="BYEEEE!":PRINTCA$"■      ";:GOTO860
370 IFC=24THEN200
380 K$="N":GOTO200
390 GOSUB770:RESTORE:W=3*O:F=0:FORM=1TO2:FORL=1TO8:READX,Y,Z
400 IFP(X)+P(Y)+P(Z)=WTHENF1=S(X):F2=S(Y):F3=S(Z):F=M
410 NEXTL:W=3*C:NEXTM:FORM=1TO2:W=0:FORL=1TO9:READX:W=W+P(X):NEXTL
420 IFF=0ANDW=180ORW=171THENF=3
430 NEXTM:RETURN
440 IFF=1THENG=O
450 IFF=2THENG=C
460 FORL=1TO5:H=32:POKEF1,H:POKEF2,H:POKEF3,H:GOSUB780
470 H=G:POKEF1,H:POKEF2,H:POKEF3,H:GOSUB800:NEXTL:RETURN
480 GOSUB770
490 CH=INT(RND(1)*100)+1
499 REM   8*CRR
500 R=INT(RND(1)*9)+1:PRINTCC$"■■■■■■■■"R
510 IFP(R)=48+RTHENCM=S(R):GOTO530
520 GOTO500
530 IFCH>LV*30+8THEN670
540 CM=0
550 IFMT>=4THEN610
560 IFMT=0THEN580
570 IFP(5)=53THENCM=S(5):MT=4:GOTO670
579 REM   8*CRR
580 R=INT(RND(1)*9)+1:PRINTCC$"■■■■■■■■"R:IFR=2ORR=4ORR=6ORR=8ANDCM=0THEN500
590 IFP(R)=48+RTHENCM=S(R):MT=MT+2:GOTO640
600 GOTO580
610 R=INT(RND(1)*9)+1
620 IFP(R)=48+RTHENCM=S(R):GOTO640
630 GOTO610
640 RESTORE:W=2*O:FORM=1TO2:FORL=1TO24:READX,Y,Z
649 REM   8*CRR
650 IFP(X)+P(Y)=WANDP(Z)=48+ZTHENPRINTCC$"■■■■■■■■"Z:CM=S(Z)
660 NEXTL:RESTORE:W=2*C:NEXTM
670 POKECM,C:FORL=0TO1:POKECM+CR,0:GOSUB780:POKECM+CR,1:GOSUB800:NEXT:RETURN
679 REM CLR-  9*CRD
680 PRINT"■■■■■■■■■■"
689 REM RVS
690 PRINTTAB(7+TB)"◢      ◣"
700 PRINTTAB(7+TB)"┌──────┐"
710 PRINTTAB(7+TB)"| 1 |2 |3 |"
720 PRINTTAB(7+TB)"| ─┼─┼─ |"
730 PRINTTAB(7+TB)"| 4 |5 |6 |"
740 PRINTTAB(7+TB)"| ─┼─┼─ |"
750 PRINTTAB(7+TB)"| 7 |8 |9 |"
759 REM HOM
760 PRINTTAB(7+TB)"└──────■";
770 FORL=1TO9:P(L)=PEEK(S(L)):NEXT:RETURN
780 IFSR=1024THENPOKECR+4,17:GOSUB800:POKECR+4,16:RETURN
790 POKE36878,15:GOSUB800:GOSUB800:POKE36878,0:RETURN
800 FORD=0TO80:NEXT:RETURN
810 IFSR=1024THENFORL=CRTOCR+24:POKEL,0:NEXT
820 RETURN
829 REM HOM-  4*CRD WHT
830 FORL=0TO2:GOSUB800:PRINT"■■■■■"CA$"■INPUT NOT ACCEPTABLE":GOSUB780:GOSUB800
839 REM HOM-  4*CRD
840 PRINTCU$"■■■■■"CA$"                         ":GOSUB780:GOSUB800:NEXT
850 FORL=0TO6:GOSUB800:NEXT:RETURN
859 REM BLU
860 FORL=1TO7:PRINTMID$(K$,L,1);:GOSUB780:GOSUB800:NEXT:GOSUB810:PRINT"■":END
870 DATA1,2,3,4,5,6,7,8,9,1,4,7,2,5,8,3,6,9,1,5,9,3,5,7
880 DATA1,3,2,4,6,5,7,9,8,1,7,4,2,8,5,3,9,6,1,9,5,3,7,5
890 DATA2,3,1,5,6,4,8,9,7,4,7,1,5,8,2,6,9,3,5,9,1,5,7,3
READY.
```



Runecaster is shrouded in mystery this month.

IN THE PAST TWO MONTHS WE HAVE dealt with mapping and mazes; this month it is the turn of the third 'm' — mysteries. A mystery is a puzzle, and that (or rather a series of them) is what adventure games are all about.

In some ways a good adventure game is rather like the 'Daily Telegraph' crossword — the answer is often staring you in the face all the time — you just (!) have to read the question correctly and then manipulate linked words, objects or letters until they make sense!

Of course the object of the author/programmer is to make these puzzles fairly devious...I say and *mean* 'fairly devious'. Anyone can draw up a set of puzzles that are almost impossible to solve, but a good adventure is like a good book — it has got to keep up a certain 'pace' or it fails. Some programs are a failure for this very reason. The player is not given a chance to get the feel of the game but is thrown straight into a situation that would tax the patience of a saint!

Many people have 'cut their teeth' on adventures such as **The Hobbit** by Melbourne House and **Dungeon Adventure** by Level Nine. Although these provide two different types of quest, they both allow the player a fair bit of rope when starting: yes of course you can be killed but, if you are, you learn something in the process. In both of these programs there are also alternative paths to be taken whilst you are 'learning the ropes'...you don't take three steps and die, everytime!

First steps

'Learning the ropes', what does this entail, what do you look for...how do you get past the dragon, board the boat — or whatever? First of all, look carefully at any instructions, sometimes these can be lengthy and you will not be able to assimulate all that they have to say...but read them anyway. Later, something that happens (or doesn't) may trigger off a vague memory, so plough your way right through them all. In particular lookout for any commands that they list — INVENTORY, EXAMINE, LOOK, HELP etc.

As you move around (mapping your progress of course!) take very careful note of what the location description tells you and what you find lying around. Find an axe and later some trees by a river — perhaps you could chop one down to make a bridge.

EXAMINE everything you can. Some programs only permit this if you are already holding the object, others are more tolerant. It is all too easy to assume that a simple description implies an obvious use — which you never seem able to find!

Perhaps that axe you found might yield to examination: "very blunt, the axe-head comes off in your hand" — have you already spotted a use for a short length of wood? Examine the trees and you might find a rope you could climb. Some programs try to wear you down with a short description of everything examined but persevere and don't let this put you off looking at everything as carefully as possible.

Find out as early as you can, how many items you can pick up and retain at any time. Especially look out for anything that may enable you to carry extra items — a rucksack or perhaps even a horse! Certain objects could very well be worn; in fact some items will probably have to be worn to enable you to complete the adventure. 'Wear glasses' might give you quite a different view of things! Wearing something may also allow you to carry more — check it.

Many adventures are divided into sections, each with their own set of puzzles to solve, before you can progress to further sections. More importantly, objects from previous sections are almost certainly going to be needed to solve future puzzles.

Section jumping

This brings us to one of the most common 'ploys' found in many forms in lots of adventures — 'section jumping'. This is where you are forced to leave one section with a puzzle (or puzzles) unsolved, because a clue or object needed is in another section. More often that not, the solution to the original puzzle(s) is needed to move on from this next section!

Look out for objects that may be combined with other objects (or be altered in some way) to produce a different but useful article. The more obvious example here is a vessel, bottle or helmet that may be used later to carry water/oil etc. How about a glass object that could be broken and provide a useful cutting tool (knife!)? Stones or that strange lump of metal may be transformed into hammers or axes!

Can you buy or bribe your way out of trouble? Does the vocabulary support the word GIVE. Note also that useable objects are not always listed separately; sometimes they are within the location's description and sometimes they just appear in the graphics!

Many puzzles are a play on words or on the theme of associated ideas — carnivorous lumps of jelly might prefer to eat that corpse you just passed...instead of you! An owl may very well like to be given a pair of sunglasses...on a bright sunny day. The possibilities go on and on, but it all comes back to a careful examination and appreciation of everything you find.

Playing safe

As said before in these articles SAVE your position regularly — you don't know what is going to happen just around the corner! Try to keep one set of SAVEs that represent a 'clean' path through the adventure. By all means explore everything you come to, but when you think you know how to tackle a particular section; start from your last 'clean' SAVE and complete this latest part with no mistakes and no extra moves — your future may depend upon a light staying on or food lasting, for only a certain number of moves. It's almost certain that number will not allow for your bumbling around mazes or generally exploring!

If the facility exists, check your SCORE fairly often. Although this is no certain method it sometimes gives you a clue to whether an item just picked up is going to be useful later.

If the instructions are not clear, find out what sort of vocabulary and grammar the program recognises. Most games expect a verb followed by a noun: GO SOUTH,

CLIMB TREE, BURN PARCHMENT etc. Some will allow or occasionally need an adjective: PRESS YELLOW BUTTON, GET LARGE KEY.

As games become more complex, longer sentences are becoming acceptable: GO NORTH QUIETLY, may be necessary to ensure that you are not discovered. This can give a more interesting program, but the trend could easily backfire — what was a logical and stimulating puzzle becomes a protracted test of words and their interpretation.

Even with the obvious limitations, there is a lot to be said for the limited syntax of verb and noun only (and in that order). An adventure game written in this fashion does not become a typing competition, and you are less likely to loose the flow of ideas whilst trying to type in long sentences accurately!

Check how much of a word the program actually needs — do you really have to type in FUMIGATE FARMYARD, or will FUMI FARM suffice? I know it seems wrong but again, it is quicker and allows you to concentrate on playing the game smoothly.

Finally the vocabulary itself. As I've said before, it is not possible for a program to recognise every word you may wish to type in — there is just not enough memory available. I feel that programmers should stick to fairly 'common use' words such as 'sing' and not 'chant' or 'croon'! For all this, each of us has a different 'common use vocabulary'. If you are serious about adventures and find yourself caught out for the right word — buy a Thesaurus; this is rather like a dictionary except you are given other words meaning the same thing, instead of the meaning and derivation of a word. Probably the simplest to use is The New Collins Thesaurus and is to be highly recommended.

That just about covers our 'instant adventures guide' using the three 'm's'. If you want any other area brought to light — you know our address!

Two from Mosaic

Several months ago we mentioned new adventures in the pipeline from Mosaic Publishing, we now have some of these up and running on our Commodore 64.

The Stainless Steel Rat Saves The World, is based loosely upon Harry Harrison's well-known science fiction book ofthe same name — the book (published by Sphere Books) is included with the program.

Loading the program takes about six minutes and once loaded some difficulty was encountered in not having the slightest idea of what to do: you are slipper Jim di Griz, a super thief turned cop. Your attractive wife Angelina (lethal, lovely and also a one time super-crood!) is missing — you fear the worst and rush over to Professor Coypu's laboratory...to find him just disappearing into a time helix. Perhaps Angelina has been abducted via such a helix. What now...the time helix fades from existence...the game is over.

Do you want to try again to save the world? Shakily you fumble with the 'Y' key and this time, forewarned, you do something — anything: GO HELIX, you are in a familiar looking lab with two 'time fixators' on the table in front of you...the helix fades away...

As the world is not immediately going to fall about your ears, you slip into your normal adventuring posture and EXAMINE FIXATORS — one is Coypu's, one is Jarl's: O.K. find out how to examine one of them — not much help, try GETting one of them...yes that can be done...

Now lets look around: GO NORTH, this is where you realise that you are dealing with a different type of adventure game from normal...the response was — TRY ANOTHER OBJECT...whew, ouch!

Look more carefully at the screen — block graphics picture on the top eleven lines, then a seven line window for the description, finally a five line window for your commands and some immediate answers...like TRY ANOTHER OBJECT.

Not much to learn from the screen, the fixators have been picked up but they still show on the table, so...read the instructions again. Not a great deal of help there, either. Although 'T' to travel (using time helix, once acquired), looks hopeful. How about 'H' to ask for help (if needed) — I do like the comment in the brackets!

'I' is for inventory — well at least that is clear enough. Turn page — brief notes on time travel and finally a short cryptic bit about the four time parameters ":spin, flux, resonance and phase. You will have to experiment to discover their properties and relationships" — well, thank you.

Trying 'H' for HELP doesn't seem to be a great success at this point, it always prints up SCENARIO followed by a repeat of the location description, printed out letter — which takes about 10 seconds to get to the end of the second line — just a little slow — cumbersome, one might say.

Frantic days later I've learnt the secret. I'm sure all you clever clogs had it worked out but move three, having got Coypu's fixator should have been BUILD HELIX — knew it, all the time, didn't you — Gnashhh.

Having built my very own little time helix — time and space are my oyster — no GO NORTH, SOUTH, EAST or WEST for me — just dial in SPIN, FLUX, PHASE and RESONANCE and I can go anywhere, anytime! The only trouble is I haven't yet got the hang of what alters which, when or where. Some combinations give me a chunky attractive picture and some just deliver a description and the display of my time helix controls.

The idea behind this program is interesting to say the least, albeit I'm not sure whether I have to be a mathematical genius or an avid adventurer — what does worry me is that there may be another cracker like BUILD HELIX waiting for me to miss it. The whole operation is slow in reacting but as I don't really know yet what I'm doing, that isn't as big a snag as it may be later.

I've already been instrumental in a different Eiffel Tower being built; I've visited pre-historic swamps, the far future and an Indian Pow-Wow. But, without some more obvious coherent link to all this time/space bashing I think I'm going to loose interest soon — perhaps Mosaic will produce a help sheet — BUILD HELIX indeed!

The Saga of Eric the Viking

After my time/space adventures with the Rat, it was a pleasure to slip quietly into just one time and space based on Terry Jones's book.

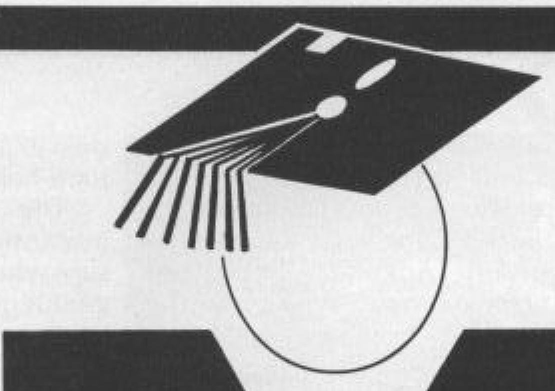
Eric returns to his village, to find that his family has been kidnapped. Your mission (should you wish to accept it) is to rescue them from their fate. You think you know they were taken away by sea — so a pretty urgent necessity must be to ready your boat and crew for a tough sea voyage.

High resolution pictures are drawn to a little over half of the screen display (the bottom nine lines are used for descriptions, actions and commands). Pictures are a little slow to complete — about 23 seconds for graphics display plus location description — but the hi-res screens are good and if you are in a hurry you may switch them off (and back on!) at will.

Objects to find and use are plentiful and the puzzles provided are logical and entertaining. Although I have not yet completed the adventure, there is a good feeling to it and I am sure that it will prove a good introduction to adventuring for many.

Mapping is of course essential and the vocabulary seems to be quite reasonable. A list of useful verbs is provided in the 22 page instruction/story booklet and shortened words are permitted. You may GET and DROP EVERYTHING and also use AGAIN to repeat a previous command. EXAMINE may be used on objects you do not have (but can see) and SAVE and RESTORE of a game position is supported. Should you get really stuck, a 'hints sheet' is available from Mosaic Publishing.

All in all a most enjoyable game even if a little slow — no pictures and the command LOOK takes about 4 seconds to display 2½lines of decription. Good plot, good graphics when you want them — buy it.



MASTERING MACHINE CODE

Arithmetic instructions are the subject of this month's installment of A.P. and D.J. Stephensons' informative machine code series.

BEFORE YOU READ ON, YOU may find it helpful to refer back to Part 1 of this series which appeared in the October issue of **Your Commodore** which dealt with binary arithmetic and two's complement.

Arithmetic, in general, implies adding, subtracting, multiplication, division and exponentiation (raising a number to a given power). All of these operations are available in BASIC, using conventional symbols which, unseen by the user, call up complex machine code subroutines buried within the interpreter ROM. The situation facing those who are more independent and not content to rely on such spoon feeding — and that means us — is less rosy. We can increment the X or Y registers, using INX or DEX and decrement them, using DEX or DEY. We can also increment or decrement the contents of any memory location by using INC or DEC respectively. Although these instructions are useful, they are little more than crude counting operations.

The 6510A microprocessor offers only two pure arithmetic instructions, ADC, which means 'ADd with Carry and SBC' which means 'SuBtract with Carry'. Because of the inherent two's complement structure, they can add or subtract any mixture of positive and negative numbers. For example, ADC can add 4 to 7, -4 to 7, -4 to -7 etc. Similarly, SBC can subtract any mixture of signed numbers. Unfortunately, ADC and SBC suffer as usual from the penalties imposed by the 6510A and indeed any other 8-bit microprocessor — the restriction on the size of numbers. Using twos complement, the largest positive number is 0111 1111 and the largest negative number is 1000 0000 which is +127 and -128 respectively. Does this mean that any arithmetic total must not exceed these limits? As far as the microprocessor is concerned, the answer is yes but, as we shall see later, there is a way of getting round the problem by storing intermediate results and sharing the work load over two or more memory locations. By employing a little ingenuity and craft, large numbers can be built up by *instalments.* In the meantime, the two instructions must be discussed in detail along with the addressing modes available.

Adding two numbers together

First, a precise definition of ADC:

ADC: Add the number, as defined by the operand, to the existing contents of the Accumulator, taking into consideration the state of the carry bit.

Expressing this is symbolic form, A = A+M+C, where M is the number defined by the operand and C is the carry bit. The operation on the right of the "=" is what happens and the left hand side shows where the result is left. This method of explaining the action of a computer instruction is called *operational symbolism.* Note that one of the numbers (represented by M in the above) will normally come from a memory location, the phrase' defined by the operand' implies this. But there is no doubt that the Accumulator is the prima donna — it is the only register in the 6510A which is equipped to handle addition or subtraction. It is no good trying to add numbers to the X or Y registers. Neither will you have any success in trying to add together two numbers in memory locations without invoking the accumulator as an intermediary. In fact, this is an ideal problem to illustrate the procedures.

Assume one number is in address $C200 and the other in $C201 and we wish to add them up and store the result in $C202.

Method 1

```
LDA $C200      (load first number)
CLC            (clear C bit)
ADC $C202      (add the second number)
STA $C202      (store result)
```

Method 2

```
LDA # 00       (clear A)
CLC            (clear C bit)
ADC $C200      (add first number)
ADC $C201      (add second number)
STA $C202      (store result)
```

The first method is one instruction shorter because the accumulator is automatically cleared of any previous contents by overwriting it with an LDA. The second method first clears the Accumulator to zero by using LDA with immediate addressing. Strangely, the 6510A has no instructions to directly clear the Accumulator, or indeed any other of the registers, so instead we have to load them with zero. Method two, although wasting one instruction, is better and tidier (?) when a number of additions have to be carried out by means of a loop.

Since the carry (the C-bit) is always lurking in the background, and may have been left in the C=1 state by a previous instruction, it is essential that it be cleared to zero, by use of CLC, before starting an addition from scratch. We should mention that there will be times, such as in multiple precision work, (discussed later) when the C-

bit is required but in simple cases, CLC must be used immediately before the use of an ADC instruction. A significant percentage of bugs in a machine code program are caused by the intrusion of the C-bit for which the programmer had forgotten to allow.

Overflow status

If, after an arithmetical instruction, the microprocessor senses that the result in the Accumulator will be outside the limits of +127 or −128 it will set the V-bit to 1 in the status register. In other words, the machine will 'warn' the programmer that the last result caused overflow status, in other words, the result is utter rubbish. Should the programmer be concerned about this? Remarkable as it sounds, the answer is — probably not! It all depends on the interpretation the programmer places on the result. If he/she interprets the 8-bit result as a signed integer, within the normal range of +127 and −128, overflow status is serious and steps must be taken in the program to initiate alternative action if it occurs. This can easily be done by using a BVS or BVC conditional branch. But, if the programmer is only concerned with absolute (unsigned) numbers, overflow status can be safely ignored because the largest number in an 8-bit register is 1111 1111 (255 decimal). The msb is interpreted as part of the overall magnitude rather than as a sign bit.

Carry status

We have seen that, at times, overflow status can be ignored when working with positive absolute numbers but, the question which naturally follows is, "What happens if the result is over 255?". This is where the C bit come in. Although the C-bit occupies a remote position in the status register, it can be treated as the 'ninth' bit of the Accumulator. Now, nine bits can accommodate numbers up to 511 decimal (1 1111 1111 binary) so the programmer knows that the C-bit, plus the bits already in the Accumulator, represent a valid total. In other words, the normal 8 bits in the accumulator form the 'lower byte' total and the C-bit becomes the lsb of the 'highbyte'. It is easy to include a conditional branch, using either BCS or BCC, at the end of an addition to check the carry status. If C=1, then another memory location can be used to hold it. This brings us to the concept of double length arithmetic.

Double length arithmetic

To add up a lot of numbers which we expect will overstretch the capacity of a single byte, we can use 'double-length' techniques. One memory location can be used to hold the low byte end of the result and another location to hold the 'high byte' end. The C-bit can act as continuity between the two.

Figure 5.1 may help in visualising double length ideas. To illustrate the technique, we will add two single length numbers together and store the result in two adjacent memory locations. The following coding is hardly worth keying in but at least it will show one of the many possible methods of handling numbers larger than 255. Assume that one number is in $C200, the other in $C201 and the double length result is to be stored in $C401 (low byte) and $C402 (high byte) addresses. The flowchart in Figure 5.2 shows the logic behind the coding

Note the following points:

(a) The first number was added to an empty Accumulator with the C-bit cleared.

(b) The second number was then added to form the low byte result. If the result exceeded 255, the C bit will have been set to 1.

(c) The high byte result is either

```
LDA # 00              (clear A)
STA $C401             (clear for low byte result)
STA $C402             (clear for high byte result)
CLC                   (clear C-bit)
ADC $C200             (add first number)
ADC$C201              (add second number)
STA $C400             (store low byte result)
BCC FINISH
INC $C402             (add 1 to high byte result)
FINISH----------
```

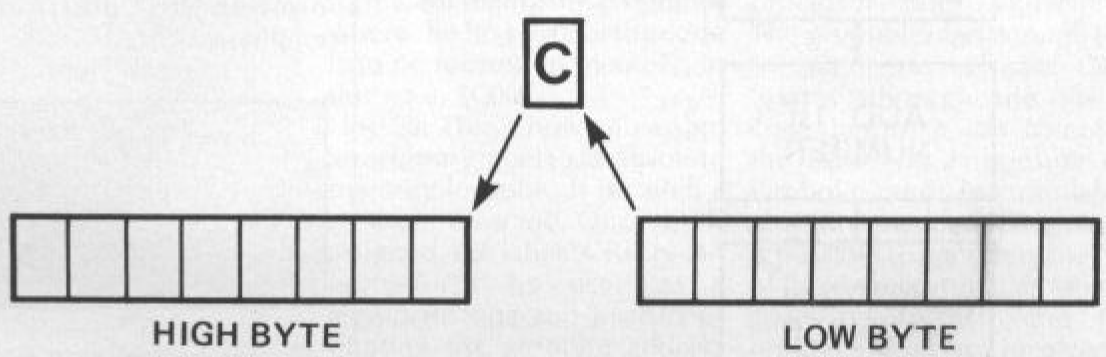
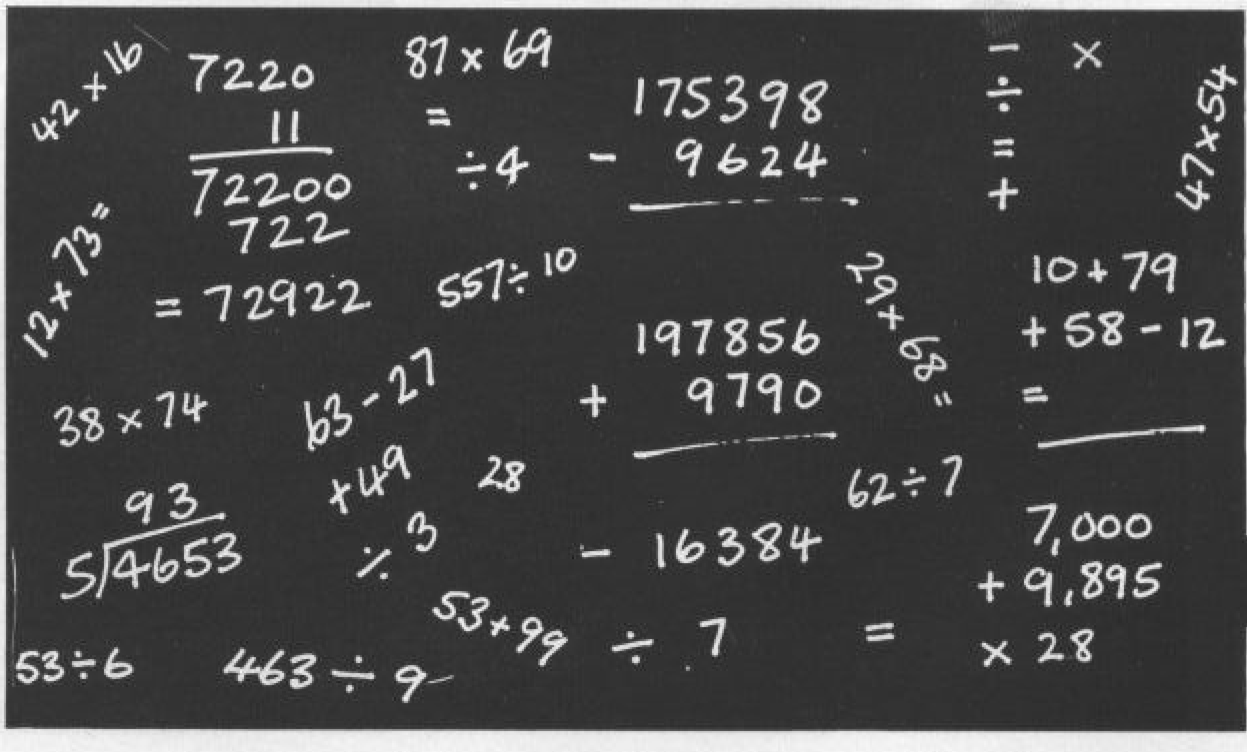


Figure 5.1



left empty if C bit was 0 or incremented if the C bit was 1.

To illustrate what actually happens at the binary level, let the original two numbers be 255 and 3 respectively:

```
   1111 1111 (255)
   0000 0011 (3)
1  0000 0010

0000  0001      0000 0010
high  byte       low byte
```

Multiple precision

When two locations (16 bits in all) are used, the maximum absolute numbers is increased to 65,535 decimal. This number, although large, may still not be enough to cope with every contingency. It should be clear that if the C-bit can be used to provide continuity between the first and second bytes there is no reason why it cannot be used to connect the second with a third byte. Three byte numbers can reach up to 2 24-1 which in decimal, is nearly 17 million. In theory, the method can be extended indefinitely to allow numbers of any magnitude and precision to be handled.

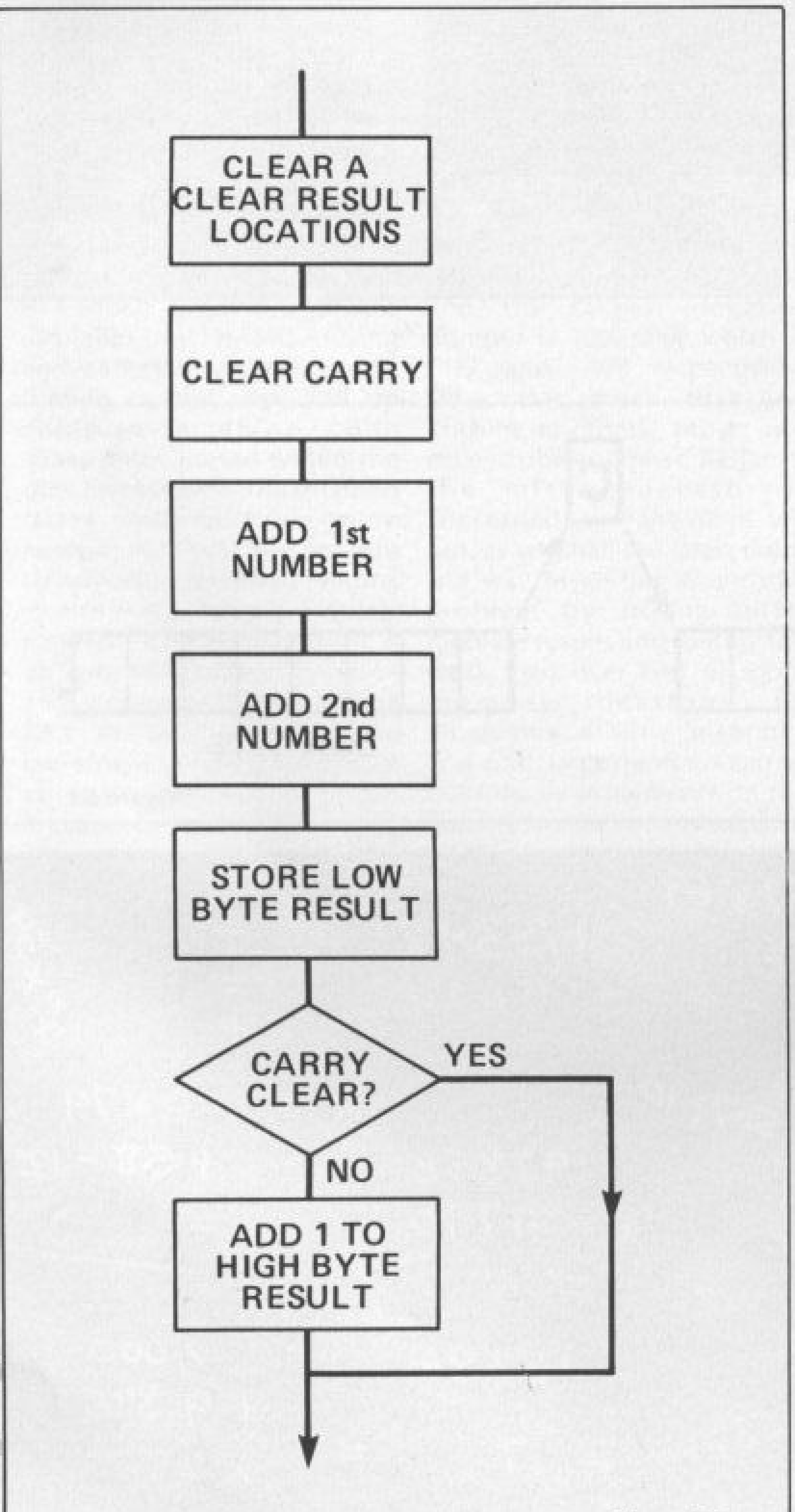


Figure 5.2

Subtracting one number from another

As before, we must start with a precise definition of SBC:

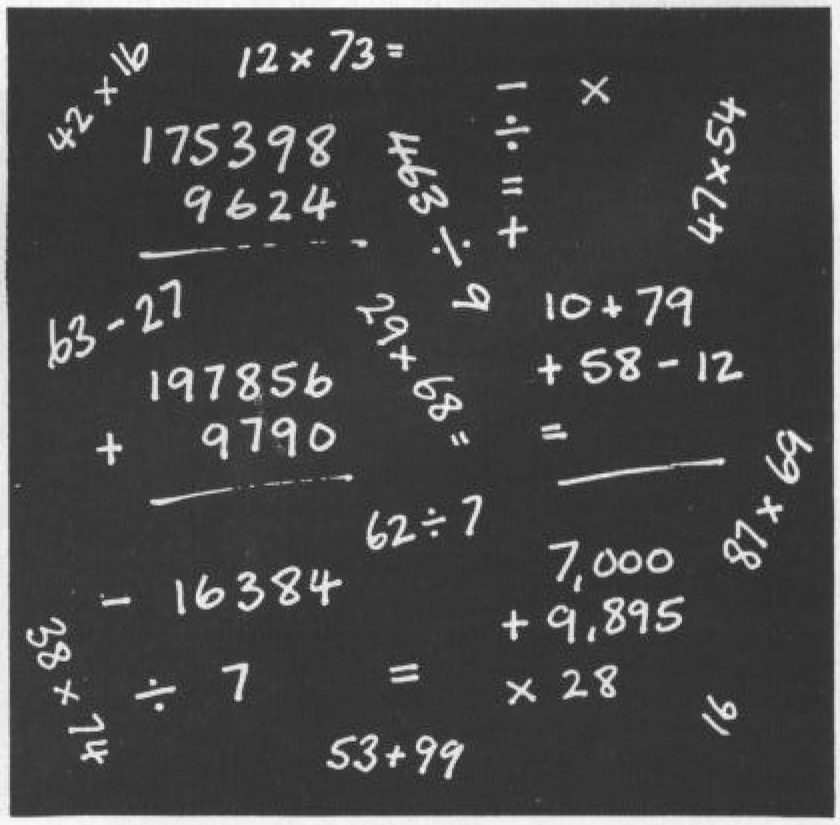
SBC: Subtract the number, as defined by the operand, from the existing contents of the Accumulator, taking into account the state of the carry bit

Unlike addition, the order in which the numbers are subtracted is important. Adding 6 to 3 is the same as adding 3 to 6. On the other hand, subtracting 3 from 6 is certainly not the same as subtracting 6 from 3. This may seen a trivial thing to bring up but it is so easy to code subtraction the wrong way round unless you remember that the subtraction is FROM the Accumulator. There is another pitfall with subtraction processes because of the way in which the C bit has to be handled. This pitfall is far from obvious. In fact it is bizzare enough to merit extra emphasis:

Before using SBC, don't clear the C bit. Instead, set it to 1 by using SEC.

The reason for this is tied up with the theory of two's complement arithmetic. Although SBC is provided, it is not essential to use it because subtracting B from A can be achieved by adding -B to A. Now, because a negative number is simply the two's complement of the equivalent positive number every bit is reversed, (flipped) including the carry bit! From this, we can deduce that although we clear the carry to zero before adding, we must *set* the carry to 1 before subtracting. This can also be justified by considering that during an addition, we 'carry' a bit over to the next significant place but, during a subtraction, we borrow a bit from it and we can't borrow a bit unless it is already a '1'. From this, it appears that the carry, in a subtraction process, should really be called the borrow. The following few lines show the procedure for subtracting the contents of \$C200 from \$C201.

```
LDA $C201
SEC               (set carry bit to 1)
SBC $C200
STA $C201         (replace result)
```



Instruction codes and addressing modes.

The following table lists the codes and addressing modes of the arithmetic op-codes.

Address type	Assembler	Hex code
Zero page	DEC $xx	C6 xx
Absolute	DEC $xxxx	CE xx xx
Zero page,X	DEC $xx,X	D6 xx
Absolute,X	DEC $xxxx,X	DE xx xx
Zero page	INC $xx	E6 xx
Absolute	INC $xxxx	EE xx xx
Zero page,X	INC $xx,X	F6 xx
Absolute,X	INC $xxxx,X	FE xx xx
Implied	DEX	CA
Implied	DEY	88
Immediate	ADC # xx	69 xx
Zero page	ADC $xx	65 xx
Absolute	ADC $xxxx	6Dxx xx
Zero page,X	ADC $xx,X	75 xx
Absolute X	ADC $xxxx,X	7Dxx xx
Absolute Y	ADC $xxxx,Y	79 xx xx
(Indirect,X)	ADC $(xx,X)	61 xx
(Indirect),Y	ADC $(xx),Y	71 xx
Immediate	SBC # xx	E9 xx
Zero page	SBC $xx	E5 xx
Absolute	SBC $xxxx	EDxx xx
Zero page,X	SBC $xx,X	F5 xx
Absolute,X	SBC $xxxx,X	FDxx xx
Absolute,Y	SBC $xxxx,Y	F9 xx xx
(Indirect,X)	SBC $(xx,X)	E1 xx
(Indirect),Y	SBC $(xx),Y	F1 xx

Status register flags

It is important to know which flags are affected, if any, after an instruction has been executed. Ignorance or forgetfulness can lead to some pretty awful programming bugs which may be difficult to trace. The table of status flags affected appeared in Table 4.2 in Part 3 of this series but, for convenience and because flag status is so important, they are repeated below except that only op-codes which have so far been covered have been included.

ADC and SBC updates N,Z,C and V flags.

CMP, CPX and CPY update N,X and C flags.

INX, INY, DEX, DEY, INC, DEC, LDX, LDY and LDA update the N and Z flags.

STA, STY STX and all the seven conditional branch instructions have no effect on any of the flags.

It is also worth pointing out again that the carry flag can be directly set or cleared by SEC and CLC. The overflow flag can not be directly set but can be cleared by CLV.

Using the MIKRO assembler

In the first part of this series, you were strongly advised to obtain one of the assembler cartridges on the market, in particular, the MIKRO cartridge obtainable from Supersoft. The example coding which have so far appeared in the series has been deliberately kept in low key but, the pace from now on will hot up a little. It is still possible for those who have not yet invested in an assembler to carry on POKEing the bytes into memory or by means of the loading program given in Part 3. For the benefit of those who may still be hesitating before purchasing the MIKRO, here is how a simple program is keyed in, (the source code).

```
10   ★=$C000
20   SCREEN =$0400
30              LDX # 0
40              LDA # 0
50 BACK         STA SCREEN,X
60              INX
70              BNE BACK
80              RTS
```

At this stage, don't worry about what the program is supposed to do because the important thing first is to analyse the format on a line by line basis.

First, notice that the assembler allows you to write line numbers. These are for convenience and reference only. You can't branch or jump to a line number in assembly language like you are used to in BASIC. Line 10: '★=' is the format for telling the assembler where the first byte of the code is to be located in memory, in this case, $C000.

Line 20: This is how you assign an *arbitrary* label to an absolute memory location. It is called a *symbolic operand*. Once this is assigned, the label SCREEN can henceforth be used as a legitimate operand instead of quoting the absolute address each time. This improves the readability of the code.

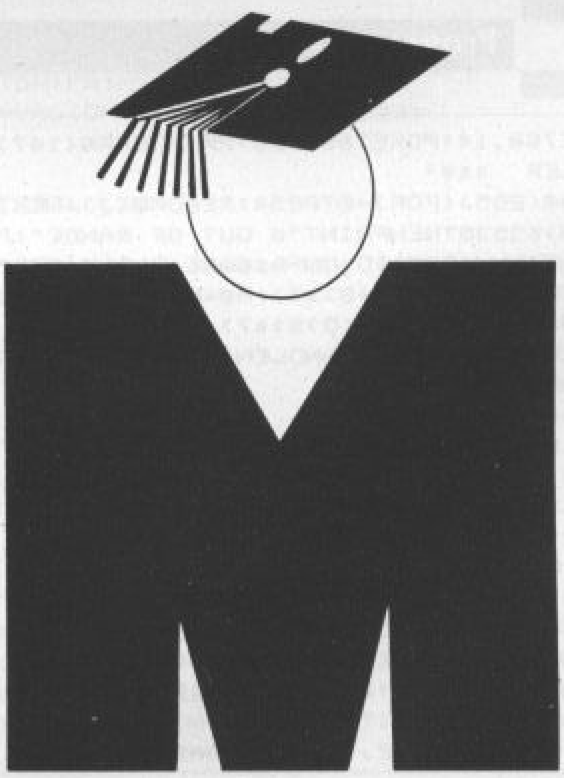
Line 50: BACK, is a *branch label* for the remainder of the line which is a simple absolute indexed instruction with a symbolic operand.

Line 70: This shows one of the most useful aspects of an assembler, the ability to use branch labels instead of laboriously working out the correct relative-address number.

Line 80: RTS (which we have not yet covered) is RETurn from Subroutine, used to steer the program back to BASIC.

Note the program is spaced out well. This can be done by the programmer or, more easily by the special assembly 'pseudo op-code' FORMAT. The only rule is that at least one space must be left between each component of the coding.

The assembler will translate the program from the original *source code*, which the microprocessor does not understand, to pure machine code, which the microprocessor does understand. The assembler has a tough job. In fact it has to make three 'passes' through the source code before it can complete the task. This is because the symbolic and branch labels must first be decoded before it is possible to complete the rest of the task. Fortunately, all this goes on unseen and the programmer is not involved.



VIC DISASSEMBLER

Here's a handy utility from John Gundry for all VIC 20 users.

THIS VIC DISASSEMBLER should be of interest to anyone following our excellent Machine Code series. It just fits into the unexpanded VIC, but will probably run on most Commodore machines with little or no modification. There are five commands the program will recognise:

?Dnnnn	Calculates and prints out the denary equivalent of the four digit hexadecimal number nnnn
?Hn	Calculates and prints out the hexadecimal equivalent of the denary number n. Where n is an integer in the range 0 to 65535.
ON	Enables output to the printer.
OFF	Disables output to the printer.
IDSnnnn	Disassembles memory from address nnnn, where nnnn is a four digit hexadecimal number. Use shift to pause the listing, or CTRL to return to INPUT mode.

The printer is assumed to be device 6; this can easily be amended in LINE 42 to suit the particular printer being used.

The part of the program which will probably need to be modified to enable the program to run on other machines are the two POKES in line number one. These set the Error Message Link vector to an RTS instruction in ROM. A crude but effective way to stop the generation of unwanted ERROR messages. Please note these POKEs must be left out until you have a fully working copy of the program.

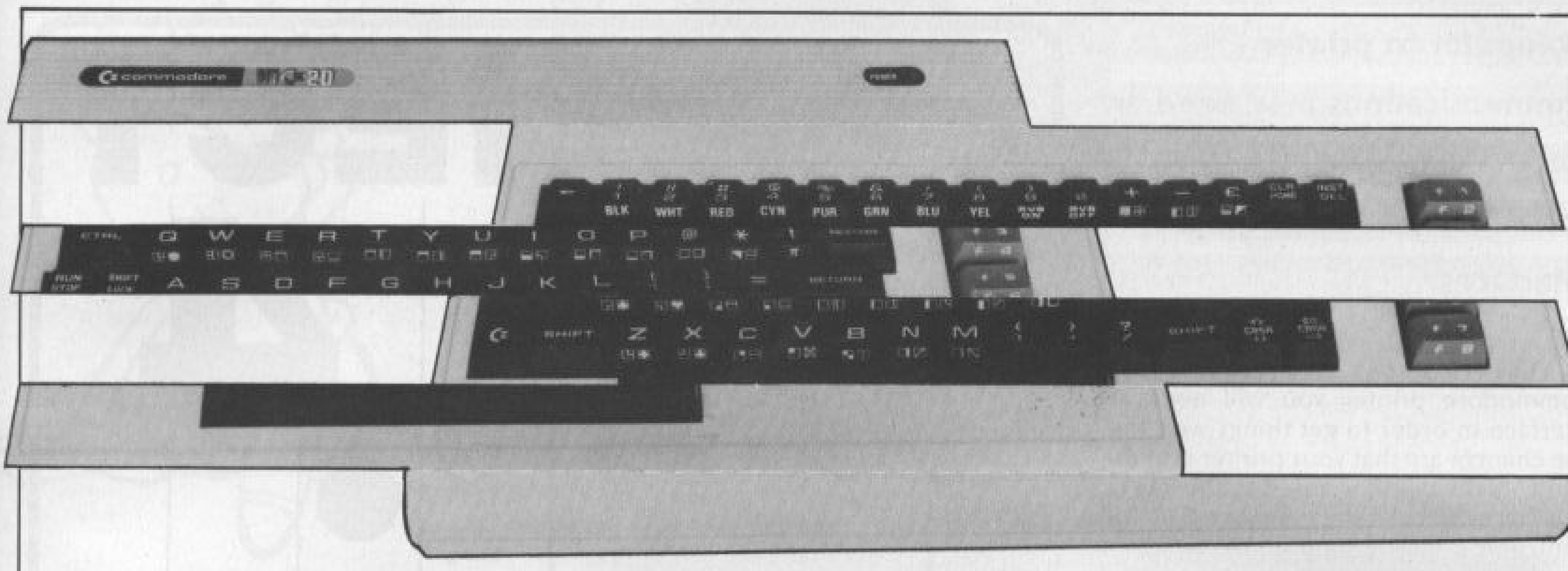
The generation of a blank line after any JMP, JMP (1) or RTS instruction is an innovation which greatly improves the readability of the disassembled code.

	Function
2	Dimensions A$(n) and READS pseudo 6502 mnemonics into it.
3-8	Subroutine to convert denary number to hexadecimal.
9-11	Subroutine to convert hexadecimal number to denary.
12-31	Main routine for disassembly.
32-38	Decodes the pseudo 6502 mnemonics to produce a more standard format for the disassembly.
39-46	Data for the pseudo 6502 mnemonics.

	Main Variables
A&(n)	Psuedo 6502 mnemonics. The opcodes correspond to the array subscripts.
DN	Denary number for conversion to hexadecimal
H&	Hexadecimal number.
M&	One line of disassembly.
CM&	Command.
S	Holds address for disassembly
PR	Printer flag. PR=1 means output to printer and screen. PR=0 means just screen.
VAL A&(n)	Number of BYTES used by instruction.

Program Listing

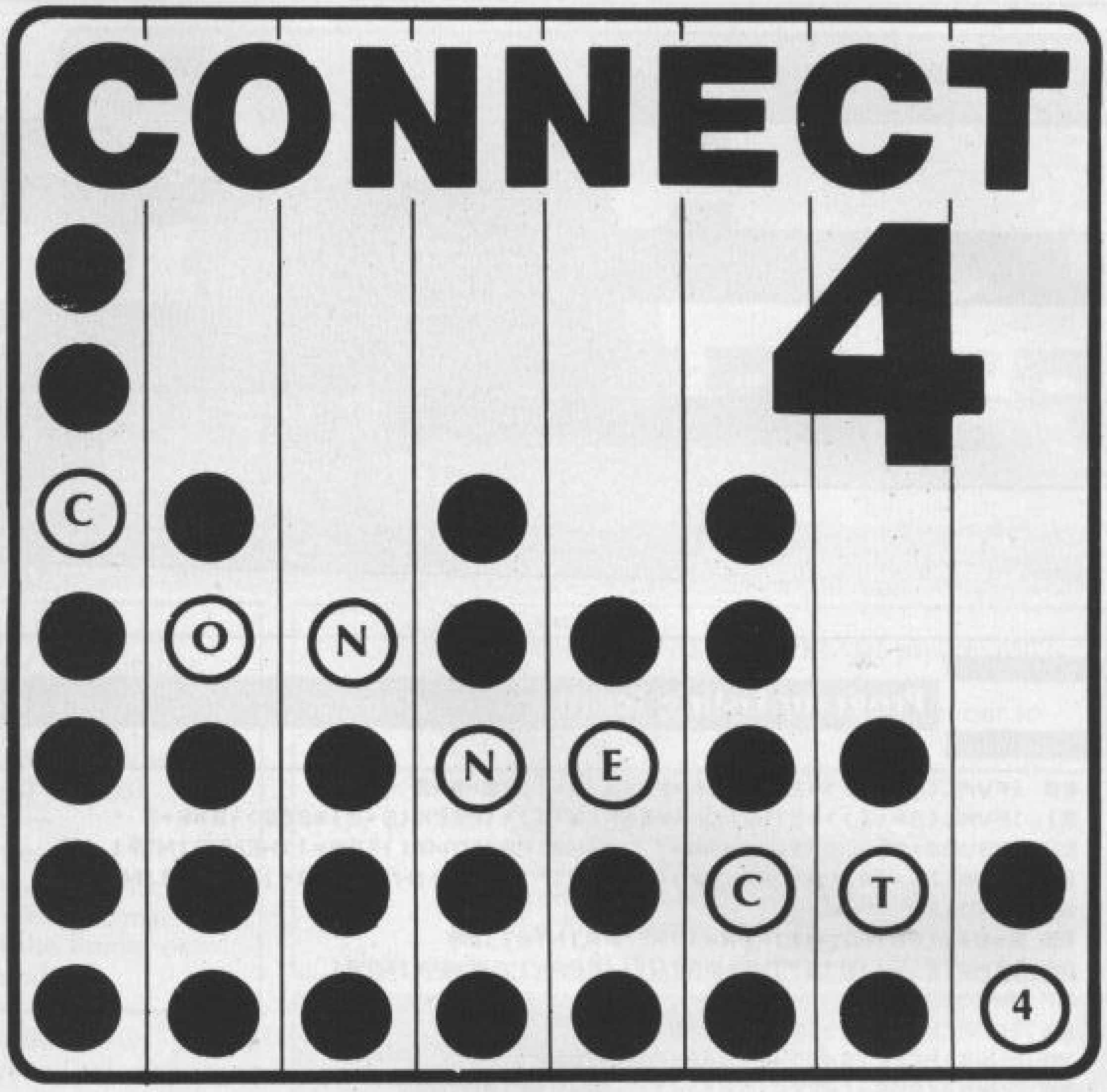
```
1 POKE768,14:POKE769,225:PRINTCHR$(147);"****  CBM VIC 20  ****":PRINT"***  DISA
SSEMBLER  ***"
2 DIMA$(255):FORJ=0TO254:READA$(J):NEXT:GOTO39
3 IFDN>65535THENPRINT"# OUT OF RANGE":PRINT" ERROR":PRINT:GOTO40
4 A=INT(DN/4096):D=DN-A*4096:B=INT(D/256):D=D-B*256:C=INT(D/16):D=D-C*16
5 H$=CHR$(A+48-(A>9)*7):H$=H$+CHR$(B+48-(B>9)*7):H$=H$+CHR$(C+48-(C>9)*7)
6 H$=H$+CHR$(D+48-(D>9)*7)
7 IFLEFT$(H$,1)="0"ANDLEN(H$)>2THENH$=RIGHT$(H$,LEN(H$)-1):GOTO7
8 RETURN
9 HE$=CM$:E=0:DN=0
10 IFLEN(HE$)<>4THEN40
11 FORI=3TO0STEP-1:DN=DN+(ASC(MID$(HE$,4-I,1))-48+7*(ASC(MID$(HE$,4-I,1))>57))*1
6↑I:NEXT:RETURN
12 M$=".,":P=PEEK(653):IFPAND1THEN12
13 IFPAND4THEN40
14 IFS>=65535THEN40
15 DN=S:GOSUB3:GOSUB37
16 IFLEN(H$)<4THENFORJ=LEN(H$)TO3:H$="0"+H$:NEXT
17 M$=M$+H$+"   ":Z=PEEK(S):IFA$(Z)=""THEN31
18 IFVAL(A$(Z))=1THENM$=M$+MID$(A$(Z),2,3):GOTO25
19 IFMID$(A$(Z),2,1)="B"ANDA$(Z)<>"1BRK"THEN28
```



Program Listing (cont.)

```
20 IFVAL(A$(Z))=2THENDN=PEEK(S+1):S=S+2
21 IFVAL(A$(Z))=3THENDN=PEEK(S+1)+(PEEK(S+2)*256):S=S+3
22 GOSUB3:GOSUB32:M$=M$+"   "+H$:PRINTM$:IFPR=1THENPRINT#1,M$
23 IFA$(Z)="3JMP"ORA$(Z)="2JMPI"THENPRINT:IFPR=1THENPRINT#1
24 GOTO12
25 S=S+1:PRINTM$:IFPR=1THENPRINT#1,M$
26 IFA$(Z)="1RTS"THENPRINT:IFPR=1THENPRINT#1
27 GOTO12
28 IFPEEK(S+1)>127THEN30
29 DN=S+PEEK(S+1)+2:S=S+2:GOTO22
30 DN=S-(255-(PEEK(S+1)+1)):S=S+2:GOTO22
31 M$=M$+"???":GOTO25
32 H$="$"+H$:IFMID$(A$(Z),5)="I"THENH$="("+H$+")":GOTO36
33 IFMID$(A$(Z),5)="#"THENH$="#"+H$:GOTO36
34 IFMID$(A$(Z),5,1)="("THENH$="("+H$+"),"+MID$(A$(Z),6):GOTO36
35 IFMID$(A$(Z),5)="Y"ORMID$(A$(Z),5)="X"THENH$=H$+","+MID$(A$(Z),5)
36 M$=M$+MID$(A$(Z),2,3):RETURN
37 IFLEN(H$)<4THENFORJ=LEN(H$)TO3:H$="0"+H$:NEXT
38 RETURN
39 OPEN2,0
40 CM$="":PRINT"READY.":INPUT#2,CM$:PRINT:PRINT
41 IFLEFT$(CM$,3)="DIS"THENCM$=RIGHT$(CM$,4):GOSUB8:S=DN:GOTO12
42 IFCM$="ON"ANDPR=0THENPR=1:OPEN1,6:GOTO40
43 IFCM$="OFF"ANDPR=1THENPR=0:PRINT#1:CLOSE1:GOTO40
44 IFLEFT$(CM$,2)="?D"THENCM$=RIGHT$(CM$,4):GOSUB8:PRINTDN:PRINT:GOTO40
45 IFLEFT$(CM$,2)="?H"THENCM$=RIGHT$(CM$,LEN(CM$)-2):DN=VAL(CM$):GOSUB3:GOSUB37:
PRINTH$:PRINT:GOTO40
46 PRINT"?SYNTAX":PRINT" ERROR":GOTO40
47 DATA1BRK,2ORA(X,,,,2ORA,2ASL,,1PHP,2ORA#,1ASL,,,3ORA,3ASL,,2BPL
48 DATA2ORA(Y,,,,2ORAX,2ASLX,,1CLC,3ORAY,,,,3ORAX
49 DATA3ASLX,,3JSR,2AND(X,,,2BIT,2AND,2ROL,,1PLP,2AND#,1ROL,,3BIT,3AND,3ROL,,2BM
I
50 DATA2AND(Y,,,,2ANDX,2ROLX,,1SEC,3ANDY,,,,3ANDX
51 DATA3ROLX,,1RTI,2EOR(X,,,,2EOR,2LSR,,1PHA,2EOR#,1LSR,,3JMP
52 DATA3EOR,3LSR,,2BVC,2EOR(Y,,,,2EORX,2LSRX,,1CLI,3EORY,,,,3EORX
53 DATA3LSRX,,1RTS,2ADC(X,,,,2ADC,2ROR,,1PLA,2ADC#,1ROR,,2JMPI
54 DATA3ADC,3ROR,,2BVS,2ADC(Y,,,,2ADCX,2RORX,,1SEI,3ADCY,,,,3ADCX
55 DATA3ORX,,,2STA(X,,,2STY,2STA,2STX,,1DEY,,1TXA,,3STY,3STA
56 DATA3STX,,2BCC,2STA(Y,,,2STYX,2STAX,2STXY,,1TYA,3STAY,1TXS,,,3STAX,,,2LDY#
57 DATA2LDA(X,2LDX#,,2LDY,2LDA,2LDX,,1TAY
58 DATA2LDA#,1TAX,,3LDY,3LDA,3LDX,,2BCS
59 DATA2LDA(Y,,,2LDYX,2LDAX,2LDXY,,1CLV,3LDAY,1TSX,,3LDYX,3LDAX,3LDXY,,2CPY#
60 DATA2CMPX,,,2CPY,2CMP,2DEC,,1INY,2CMP#,1DEX,,3CPY,3CMP,3DEC,,2BNE
61 DATA2CMP(Y,,,,2CMPX,2DECX,,1CLD,3CMPY,,,,3CMPX,3DECX,,2CPX#
62 DATA2SBC(X,,,2CPX,2SBC,2INC,,1INX,2SBC#,1NOP,,3CPX,3SBC,3INC,,2BEQ
63 DATA2SBC(Y,,,,2SBCX,2INCX,,1SED,3SBCY,,,,3SBCX,3INCX
READY.
```

Computer to printer communications may need an intermediary. David Crisp considers four Centronics interfaces.



IF YOU HAVE A 64 AND A NON-Commodore printer you will need an interface in order to get things working. The chances are that your printer is of the Centronics type as opposed to the RS232.

I have been trying out a few of the Centronic's interfaces that are available from the cheap to the expensive and most users will find one here that will be the best for their particular needs.

If you have a dot matrix printer it is possible that you could, in theory at least, print out Commodore's full character set. The simpler interfaces such as the Stack will not support this but the costlier ones will. Some interfaces will enable your printer to completely emulate a Commodore printer. They do cost a little more but it is money well spent.

Stack Centronics interface

The Stack interface consists of just a lead and tape based software to act as a driver. Used on its own the lead works with most of the commercial software which requires only a connecting lead. To use it in programs which you have written yourself you will first need to load the interface software before you load the program. Alternatively, to make the program a little easier to use, it is possible to load it from within your program. This is done by having a line in your program saying something along the lines of — 10 IF A = 0 THEN A=1:LOAD "DRIVER NAME",8,1 :rem. If you have a tape based set up change 8,1 to 1,1. When your program is run it will load software and then continue to run.

The advantage with the Stack interface is that as the code is relocatable you should be able to find an area of RAM which is unused and tuck it away there. Routines are built into the software so that it can be saved and loaded into a particular part of RAM. No help was given in the manual regarding saving the tape based software to disc, but the manual implies that a disc based version is available. Unfortunately there is no indication on the box it arrived in to say whether it is disc or tape based. This is one of the cheaper interfaces available although it is still fairly expensive. However, the quality of the components used in the lead is very high. The cable itself is a thick round type as opposed to ribbon and the plugs at each end are chunky and solid.

It does what it is supposed to do, but it is not the most convenient of interfaces to use, and for what it consists of I feel it is overpriced

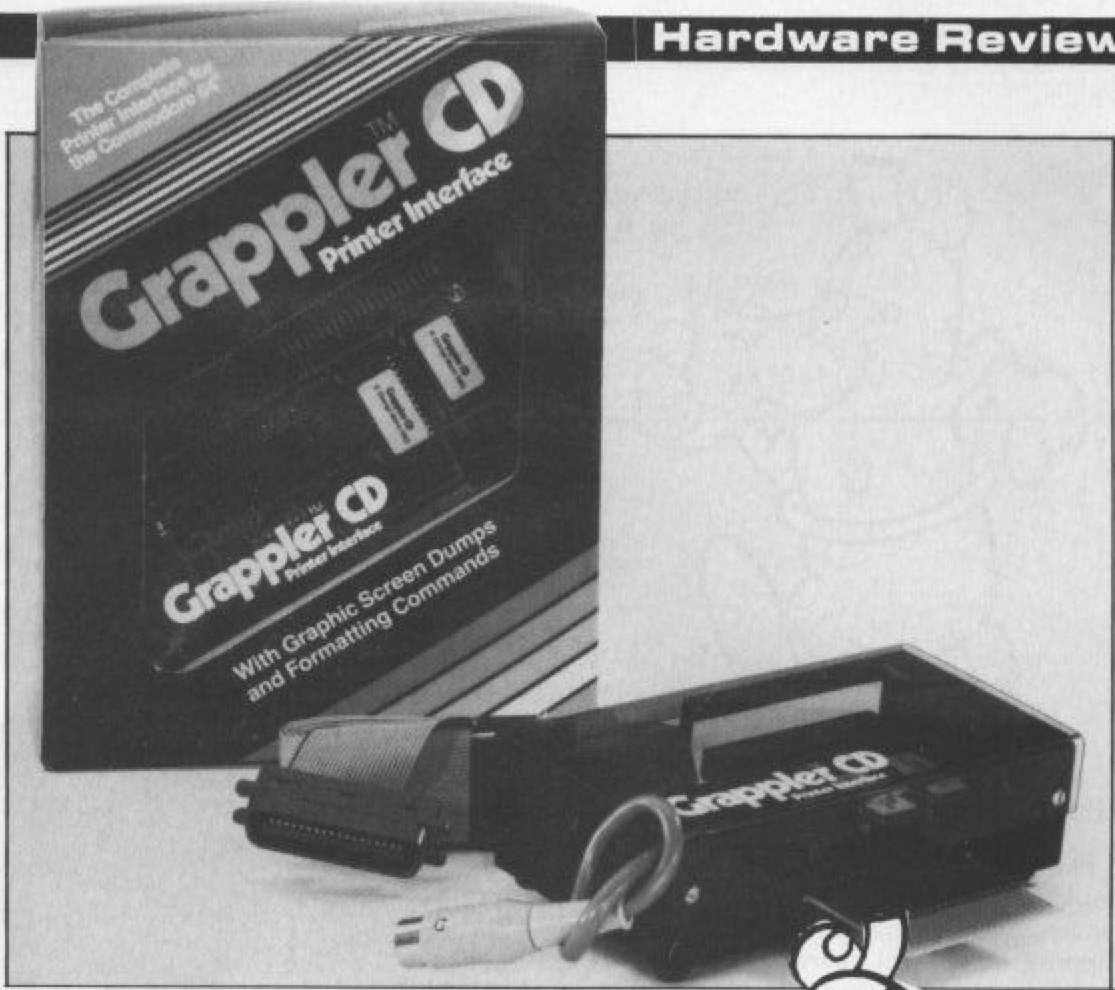


RAM Electronics

The RAM electronics interface goes by the name of Trippler. It is a black box with two leads. One lead goes from the box to the serial port, the other to your printer. The whole assembly then plugs into the user port of the 64. One thing that cannot be criticised on the Trippler is the length of the lead. On the one I received there was over one yard of ribbon cable. This makes a pleasant change as often you find that the lead is just that little bit too short. The manual, if it can be described as that, is simply a sheet of A4 printed on both sides. The Trippler is not a sophisticated interface so the documentation would seem adequate.

Because all the decoding software etc. is in the black box it will not conflict with any software in RAM. This is an advantage over the Stack. The Trippler is plugged into the serial port, so if a piece of software asks which printer you have, you respond with Commodore. This type of interface is nice as it means that any software which supports a Commodore printer will run. Using it in your own programs is also very simple as it runs like a Commodore printer. The instructions are clear and explain simply all you need to know. Again this one will not support the full Commodore character set but is a good general purpose interface. For its price it is a reasonable buy; the only disadvantage is that it does require the use of the user port. There is not a through port on the Trippler, so if you wanted to piggy back peripherals the Trippler would need to go last. How this would affect the performance I do not know as none of the other things I use on my 64 have a through





port either. So far then the best interface seems to be the Trippler with less conflict in memory, ease of use and no need to load an external software driver.

To emulate or not

As you see, the above interfaces do not fully emulate Commodore printers. The next two enable some of the more popular printers to do complete emulation. With interfaces like these your choice of printer widens enormously as both these allow you to configure the interface according to printer type. In both cases the main configuration is done with dip-switches. They do cost more but with one of these you will have a very powerful printer.

Grappler

The Grappler has a smoked perspex cover which enables you to see exactly what is inside. This reveals an 8 switch dip switch.

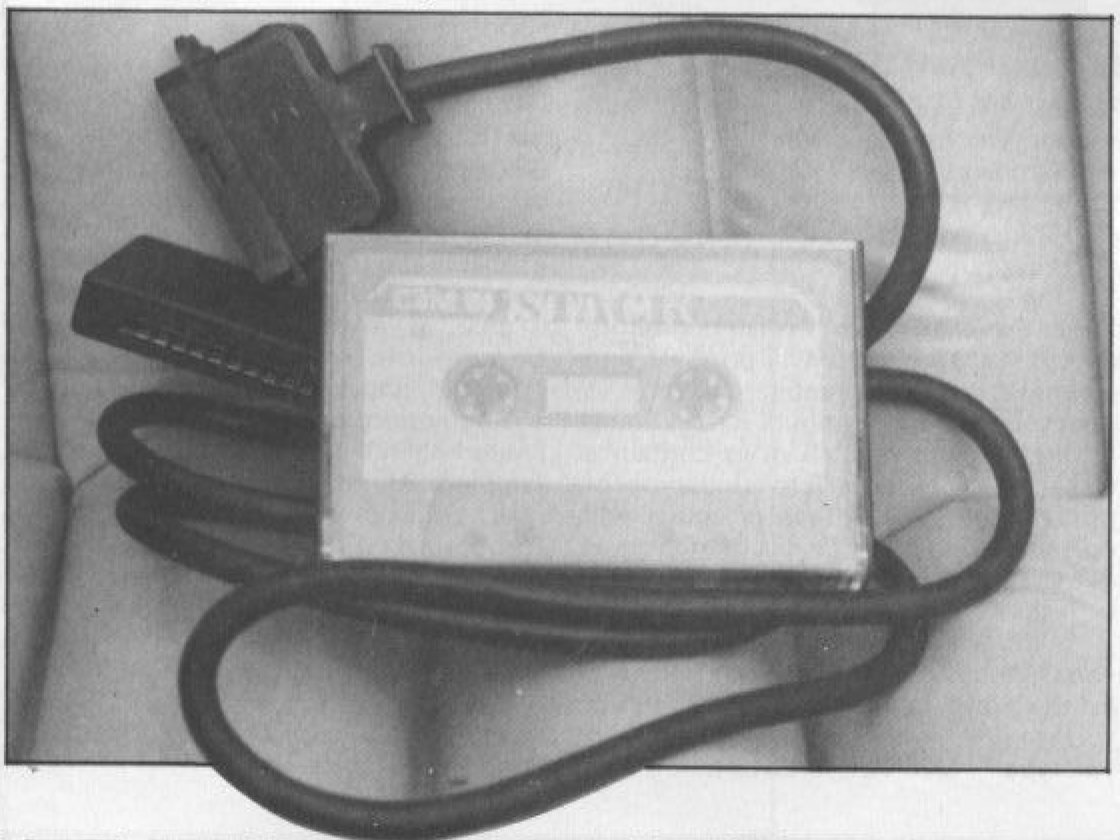
There was plenty of ribbon lead to reach the printer but the plug which goes into the serial port is very short. If you have a ordinary 64 it is O.K. but once again if you own an SX-64 modifications are required. The main unit plugs into the user port and on this one there is a through port so it is possible to piggy back. Another lead plugs into the serial port. There is another serial socket on the interface so again peripherals using the serial port can be piggy backed. Of course if you do piggy back then make sure that the Grappler comes before the printer.

Connections made, I plugged it in and started to try this new generation interface.



The manual consists of some thirty pages and all of it is needed. It is very much an American manual and tends to be a little condescending but does at least get on with the job. Many modes are available on the Grappler from straight emulation to listing mode. Many of these modes are set through the dip-switches and others can be turned on and off through software. Used as an emulation device it performs well. All the Commodore character set is printable and used with word processing packages it performs well.

Through the dip-switch it is possible to set the CD screen function, emulation mode, program listing mode, transparent mode, and text mode. These change the way the Grappler works. For instance, in listing mode instead of the difficult to decipher inverse graphics which Commodore use to show a control function, the grappler will print, between brackets a couple of letters. If the command is a cursor up then Grappler will print (csrup). This makes listings easy





to read and it is done automatically. The emulation mode would not do this conversion. Transparent mode stops any trapping of control characters. By this I mean, if a chr$(14) was sent to the printer in order to turn on a single line of double height characters this may be ignored if in emulation or listing mode. Alternatively strange print may occur. Transparent mode however allows this character to pass unchanged and you will be able to print what you like. That is an oversimplification of transparent mode, but in short, it gives total control over your printer.

One very interesting thing about the Grappler is that it contains inbuilt software to do screen dumps. This is an interesting feature and quite a useful one. Of course it will only work if your printer can do a bit image dump but it is there and it does work. However, you can only dump to the screen if the program supports screen dumps and you can break into the program and exit while the image is on the screen.

In use it worked well. To use it to its full would require some time going through the manual and unfortunately I have not been able to take it to its limits yet.

It is not cheap but it is very nearly the ultimate interface. It is of a very high quality and fairly easy to use. Whatever printer you have you should be able to configure the Grappler to suit. Most of the popular printers are mentioned in the hand book but even if it's not, you would probably find one setting that will work.

Turboprint GT

For me *this* is the ultimate interface. It is fairly simple and does not have all the functions of the Grappler, but considering the quality, ease of use, and price of the GT it is the one I think may become the most popular. It is a small black box about 5" x 3" with a ribbon connector and serial lead. On the standard version there is an extra small lead which plugs into the cassette port. This has a socket as well so you can still have the cassette plugged in. Then it simply takes five volts from the port in order to drive the GT. If you have an SX-64 you will need to cut off the cassette plug and take the 5 volts from the user port. Because it plugs into the serial port only (except for the 5V line) there is absolutely no possibility of conflict in memory. A small cutout on the top of the GT reveals four switches on a dip switch. These are used to select the device number and set the printer type. Any other selection is done under software control. It is simple, tidy and works extremely well. Like the like the Grappler will give you a listing which replaces the inverse characters with meaningful mnemonics.

Since I finished reviewing the interfaces this is the one that has remained connected. It really is excellent and represents tremendous value for money.

One last thing worth pointing out about the GT is that there is an extra slot on the bottom which will accept a printer buffer. According to the manual you should be able to get a 16 or 32K buffer but I understand that they will not be importing the 32K version. The buffer retails at the same price as the GT and this would appear to be one of the cheapest buffers I have seen.



Grappler it has various modes and sub modes. The main five are:

A. Emulation
B. Listing
C. Transparent
D. Graphics
E. Tag

These have all been explained before except tag. This mode will print a £ sign whenever an unprintable character is received. There is no inbuilt screen dump facility but the manual does contain a screen dump program for various types of printer. The screen dump of course will only work under similar circumstances as the Grappler. It is easy to change modes with the GT.

If you want a normal emulation listing you type — OPEN4,4:CMD4:LIST, this will list the program as seen on the screen. If you want to switch to listing mode then you type — OPEN4,4,10:CMD4:LIST. This

Results

If I was going out to buy an interface today I would have to say that it would be either the Turboprint GT or the Grappler. The Grappler has a few more facilities and does have through ports but taking everything into consideration I would probably aim for the Turboprint. It is hard to quantify why but I think the main thing is that as it only plugs in where a Commodore printer plugs in (5V excepted) it makes your printer closest to a Commodore printer. If my funds were limited then I would go for the RAM Trippler. All the interfaces worked, it really boils down to what you need.

Turboprint GT ..Imported by Norbain Micros Price £65.00
Grappler . . . . P & P Micro Price £136.85
RAM Trippler . . . RAM Electronics, Fleet Price £49.95
Stack 100 . . . . Stack 100 Liverpool. Price £24.00

THE 64 SOFTWARE CENTRE
1 Princeton Street, London WC1
01-430 0954

Business software pricelist — prices include VAT
d — disk c — cassette r — cartridge

Accounting systems (Office use)		
Anagram Systems — Sales Ledger	d	£75.00
Anagram Systems — Purchase/Nominal Ledger	d	£75.00
Anagram Systems — Cashbook	d	£75.00
Gemini — Cashbook (with nominal analysis)	d	£64.95
Gemini — Cashbook (with nominal analysis)	c	£59.95
Gemini — Final Accounts	d	£64.95
Gemini — Final Accounts	c	£59.95
Gemini — Cashbook/VAT/Final Accounts	c	£89.95
Ramtop — Accounts Package	d or c	£75.00
Microsimplex — Cashbook	d	£172.50
Studio — Payroll 64	d	£34.95
Dell — Invoice generator	d	£14.95
Stock Control systems		
Practicorp — Inventory 64	d	£29.95
Gemini — Stock Control	d	£29.95
Gemini — Stock Control	c	£24.95
Anagram — Stock Control	d	£75.00
Budgeting/Forecasting systems		
Adamsoft — Budgeteer	c	£8.95
Home applications		
Commodore — Magic Desk	c	£57.50
Gemini — Home Accounts	c	£19.95
Gemini — Home Accounts	d	£24.95
Fieldmaster — Home Accounts	d or c	£19.95
Adamsoft — Checkbook Manager	d	£14.95
Database Filing Systems		
Saxon — Figaro 64	d	£86.25
Handic — Diary 64	r	£29.95
Fieldmaster — Mail Label	d	£39.95
ProLine — Mailpro 64	d	£79.35
Audiogenic — Magpie	r+d	£75.00
BI — The Consultant	d	£125.00
Practicorp — Practifile 64	d	£44.50
Precision — Superbase 64	d	£89.95
Simply File	d	£69.00
Gemini — Database	d	£24.95
Gemini — Database	c	£19.95
Gemini — Mailist	d	£24.95
Gemini — Mailist	c	£19.95
Fieldmaster — Record Card	d or c	£19.95
Fieldmaster — Mail Label	d or c	£19.95
Dell — Superfile	d	£14.95
Spreadsheets		
Hesware — Multiplan 64	d	£104.95
Handic — Easy Calcresult	r	£49.95
Handic — Advanced Calcresult	r+d	£90.00
Practicorp — Practicalc 64	d	£44.50
Practicorp — Practicalc 64	c	£39.95
Supersoft — Busicalc 1	d or c	£17.95
Supersoft — Busicalc 3	d	£81.65
Fieldmaster — Worksheet	df or c	£19.95
Statistics		
Handic — Stat 64	r	£29.95
Word Processors		
B.I. — Paperclip 64	d	£98.90
Simple — Simply Write	d	£46.00
Viza — Vizawrite 64	d	£79.95
Viza — Vizawrite 64	r	£89.95
Viza — Vizawrite + Vizaspell	d	£99.95
Hesware — Heswriter	r	£39.95
Fieldmaster — Pagewriter	d or c	£29.95

Utilities		
Adamsoft — Sprite Aid	c	£6.95
Adamsoft — Quickchart	c	£6.95
Adamsoft — Screen Graphics	d	£14.95
Adamsoft — Ultrabasic	d	£19.95
Adamsoft — Ultrabasic	c	£14.95
Adamsoft — Chartpak 64	d	£24.95
Adamsoft — Cadpak 64	d	£35.00
Adamsoft — Superdisk	d	£12.50
Adamsoft — Graphics Designer	d	£19.95
Adamsoft — Zoom Pascal	d	£29.95
Audiogenic Monitor	r	£29.95
Audiogenic — Forth 64	r	£29.95
Handic — Mon 64	r	£34.95
Handic — Forth 64	r	£34.95
Handic — Graf 64	r	£29.95
Fieldmaster — Poster Printer	c	£19.95
Commodore — Simons Basic	r	£47.50
Kuma — BC Basic	r	£57.50
Commodore — Logo	d	£57.50
Commodore — Pilot	d	£75.00
Hesware — Forth 64	r	£54.50
Practicorp — 64 Doctor	d	£19.95
Supersoft — Master 64	d	£71.30
Supersoft — Victree	r	£56.35
Supersoft — Mikro Assembler	r	£59.80
Aztec — Turbo extended basic	c	£14.95
Jetpack compiler	d	£39.95
Jetpack compiler	c	£14.95
Koalapad touch tablet	r	£89.95
Quicksilva — Doodle	d	£14.95

GAMES, ADVENNTURES, SIMULATIONS
A wide range on disk, cassette, cartridge, (American and domestic)-including War and Flight Simulations.

ALL IN STOCK NOW — Callers welcome (10am-6pm incl Saturdays)

Mail Order: Cheque or PO — add £3 for items over £30, others free.
Credit cards (Access/Visa): phone or write. Immediate despatch.

The 64 Software Centre, (Dept YC)
1 Princeton Street, London WC1

Please supply the following items:

1 .... **Qty**..... **@£** ....
2 .... **Qty**..... **@£** ....
3 .... **Qty**..... **@** ....
4 .... **Qty**..... **@** ....
TOTAL £....................
POSTAGE £....................
TOTAL £....................

Name .... **Address** ....
Visa/Access Card No ....
....
Date .... **Signature** ....





A game must be more than just a mass of objects moving around the screen. In the second part of this great new series, David Rees shows how to provide a background to your game.

THIS MONTH, I INTEND TO COVER aspects of the background of the game. A background may be used either as an enhancement, or an actual part of the game, but both give an extra quality of detail to the game. For example, a 'shoot the aliens in space' game is all very well with a matt black background but the addition of a few coloured stars takes very little programming, but adds depth and more realism.

I will avoid explaining the first category any further, as it is up to your own artistic abilities. Backgrounds that take a more active roll (principally, those that get in the way of the foreground objects) are more important.

The first and most important operation concerned with a background is to let the foreground objects (moving, co-ordinated objects such as your space ship) know obstacles are there. Routine three handles this, and is explained later. This routine will make detection easy, but it is up to you to decide whether objects are simply stopped (eg. as in Pacman), or are destroyed (eg. Scramble) by obstacles.

You may also decide that changing the background is essential to the game. However, be warned that even one POKE used regularly can slow a BASIC game by a significant amount. Many arcade games use a screen scrolling feature, so I have included machine code routines to do this. Even in code, scrolling takes quite a time, so use these routines sparingly. To increase speed, you have the option of just scrolling characters, which can still be effective while halving the execution time.

Sound is another enhancement to a game. It is up to you to contrive your own ear shattering effects, but as shot sounds have to be executed fast (so as not to slow down the action), I have included a machine code gun shot routine.

As with last month's article, some machine code routines are included to help with your program. To start, the listing gives a general routine for POKEing the code to memory. They may be combined in any order (as long as you remember their start positions for the SYS command), and are independent of one another. Here is an explanation of each routine:

Routine three: This measures sprite collisions with any background object. To tell the routine whether to search for a collision with that sprite use POKE53049+ sprite no.x2, 1 for see or 0 to blind it. PEEK(53050+ sprite no.x2) gives a result for that sprite after the routine has been run. There are two other important locations, 53065 and 53066. These hold centering values for X and Y, respectively, and are normally set to eight, where the routine searches under the sprite. By changing these numbers, you can search around the sprite (eg. if 53065 contained zero, the routine would search to the left of the sprite) and see ahead of it. This is especially useful in a Pacman type game, where you have to stop sprites going through the wall.

Routines four and five: these routines scroll the screen down and to the left, respectively, and they are easy to use and are versatile. POKE 53067 determines the start line of scrolling, and 53068 controls the finish, allowing your life and score display to stay still during the action. Please note that there are no limits to these registers, so you must be careful not to put in too high a number, as your program may be scrolled. Also note that the first line/column is not cleared after scrolling. This allows many special effects, such as using any character as the screen background. Moving characters can simply be placed on the line below and to the left of the first. Finally, POKE 53069 determines whether both screens (2) or just the character screen (1) are scrolled.

Routine six: this simply produces a gunshot taking very little time, as nothing needs POKEing. Also, if you follow each sound with a channel one cut off (eg. POKEing the wave register to zero) and repeat the effect fast, you will get a machine gun effect.

If any of the above is not clear, listing 2 gives an example of how to use each of the routines, including setting the registers.

N.B. Before using listing 2, the machine code routines should already have been POKEd in by routines in listing 1.

Listing 1

```
1 REM********************
2 REM*GAME MACHINE CODE*
3 REM*SUBROUTINES PRT 2*
4 REM*         BY      *
5 REM*   DAVID REES    *
6 REM*****************
10 A(3)=13025
20 A(4)=26499:A(5)=25573:A(6)=3477
500 FORN=3TO6:T=0
510 FORM=0TO255:READA:IFA<0THEN530
520 POKE48896+N*256+M,A:T=T+A:NEXT
530 PRINT"TOTAL FOR ROUTINE"N" IS:"T
540 PRINT"IT SHOULD BE:"A(N)
550 IFA=-2THENEND
560 NEXTN
1498 REM*         ROUTINE 3      *
1499 REM*SPRITE/BCKGRD COLLISION*
1500 DATA169,1,133,251,160,,132,2
1510 DATA185,57,207,240,81,173,16,208
1520 DATA37,251,240,2,169,128,133,252
1530 DATA185,,208,24,74,24,101,252,56
1540 DATA233,20,24,109,73,207,24,74,24
1550 DATA74,24,133,252,185,1,208,24,74
1560 DATA56,233,33,24,109,74,207,24,74
1570 DATA24,74,24,168,173,136,2,133,253
1580 DATA165,252,192,,240/11,24,105,40
1590 DATA144,3,24,230,253,136,208,245
1600 DATA133,252,160,,177,252,164,2
1610 DATA153,58,207,6,251,200,200,132,2
1620 DATA192,16,208,157,96,-1
1798 REM*         ROUTINE 4      *
1799 REM*SCROLL SCREEN DOWN*
1800 DATA173,76,207,56,237,75,207,176,1
1810 DATA96,24,201,,208,1,96,170
1820 DATA172,76,207,173,136,2,133,254
1830 DATA133,252,169,,192,,240,13,24
1840 DATA105,40,144,5,24,230,254
1850 DATA230,252,136,208,243
1860 DATA141,250,207,133,253,24,105,40
1870 DATA144,3,24,230,252,133,251
1880 DATA141,252,207,173,77,207
1890 DATA141,255,207,165,254
1900 DATA41,3,24,105,216,24,141,251,207
1910 DATA165,252,41,3,24,105,216,24
1920 DATA141,253,207,142,254,207,160,39
1930 DATA177,253,145,251,136,208,249
1940 DATA177,253,145,251,165,251,56
1950 DATA233,40,176,2,198,252,24
1960 DATA133,251,165,253,56,233,40
1970 DATA176,2,198,254,24,133,253,202
1980 DATA208,216,173,250,207,133,253
1990 DATA173,251,207,133,254
2000 DATA173,252,207,133,251
2010 DATA173,253,207,133,252
2020 DATA174,254,207,206,255,207
2030 DATA173,255,207,208,185,96,-1
2098 REM*         ROUTINE 5      *
2099 REM*SCROLL SCREEN LEFT*
2100 DATA173,76,207,56,237,75,207,176,1
2110 DATA96,24,201,,208,1,96,170
2120 DATA172,75,207,173,136,2,133,254
2130 DATA133,252,169,,192,,240,13,24
2140 DATA105,40,144,5,24,230,254
2150 DATA230,252,136,208,243
2160 DATA141,250,207,133,253,24,105,1
2170 DATA144,3,24,230,252,133,251
2180 DATA141,252,207,173,77,207
2190 DATA141,255,207,165,254
2200 DATA41,3,24,105,216,24,141,251,207
2210 DATA165,252,41,3,24,105,216,24
2220 DATA141,253,207,142,254,207,160,0
2230 DATA177,251,145,253,200,192,39
2240 DATA208,247,165,251,24
2250 DATA105,40,144,3,24,230,252
2260 DATA133,251,165,253,24,105,40
2270 DATA144,3,24,230,254,133,253,202
2280 DATA208,218,173,250,207,133,253
2290 DATA173,251,207,133,254
2300 DATA173,252,207,133,251
2310 DATA173,253,207,133,252
2320 DATA174,254,207,206,255,207
2330 DATA173,255,207,208,187,96,-1
2398 REM* ROUTINE 6 *
2399 REM* GUN SHOT  *
2400 DATA169,,141,4,212,169,10
2410 DATA141,5,212,169,1,141,6,212
2420 DATA169,15,141,24,212,169,129
2430 DATA141,4,212,169,50,141,1,212
2440 DATA96,-2
```

Listing 2

```
1 REM********************
2 REM* EXAMPLES OF USE *
3 REM*       OF        *
4 REM*GAME MACHINE CODE*
5 REM*SUBROUTINES PRT 2*
6 REM*        BY       *
7 REM*   DAVID REES    *
8 REM********************
9 REM*CODE START BYTES *
10 R3=49152+2*256:R4=49152+3*256
20 R5=49152+4*256:R6=49152+5*256
28 REM*SET UP SPRITE/BCKGRD REGISTERS
29 REM*SEE/NOT SEE
30 FOR N=0 TO 7
40 POKE53049+N*2,0
50 POKE53050+N*2,0
60 NEXT
70 POKE53049,1
79 REM*CENTERING
80 POKE53065,8:POKE53066,8
97 REM*SET UP SCROLL LEFT REGISTERS
98 REM*SET UP SCROLL DOWN REGISTERS
99 REM*START LINE
100 POKE53067,1
109 REM*FINISH LINE
110 POKE53068,20
119 REM*COLOUR/NOT
120 POKE53069,1
128 REM*SET UP GUNSHOT REGISTER
129 REM*SET SHOT PITCH
130 POKER6+26,50
170 V=53248:POKEV+32,0:POKEV+33,6
180 POKEV,172:POKEV+1,140:POKEV+16,0
190 POKEV+39,14:POKEV+21,1
199 REM*[CLR HOME][L.BLUE]
200 PRINT"PEEK BELOW SPRITE IS:"
210 A$="                    "
219 REM*[GREEN]
220 PRINT"";
230 FORN=0TO87:PRINTA$;:NEXT
240 PRINT"THIS BIT IS NOT SCROLLING"
290 T=0:P=53050:S=1144+38
299 REM*[HOME]..[CRSR LEFT]
300 SYSR5:SYSR3:PRINT"",,PEEK(P)" ":GETA$
310 POKES+INT(RND(1)*17)*40,1:IFA$=""THEN300
320 SYSR6:GOTO300

READY.
```

Education is a continuous process and, with more of us using computers, Margaret Webb reviews some of the books which are trying to encourage education through computers.

Title:	The Commodore 64 Learning Tree.
Author:	Tony Noble
Publisher:	Sigma Press
Price:	£6.95

THIS BOOK'S A LITTLE PUZZLING. From the cover it appears to be a standard educational book but a quick scan within, reveals a collection of educational programs which involve a lot of typing. There are a number of introductory passages aimed at parents, teachers and children. Ignoring the inevitable amount of boring spiel, there is a useful guide to the control codes and some snippets on entering the programs. A list showing for which age group each game is suitable precedes the games.

The games themselves are a good mix and are specifically intended to help educate the player (if you only want to destroy invaders or something similar...ignore this book).

There are a nimber of games to help teach the younger age group (3-6 years) counting, simple arithmetic, simple numeric sequences and odd man out. Whilst being useful, I feel these are rather a dry lot, especially when one appreciates how difficult it can be to maintain the attention of younger children.

The games for the 7 to 14 year olds are a lot more varied and interesting and include searching games using coordinates, logic, word games, homonyms and some arithmetic. For the older children, there were games testing French and Geography.

On the whole, the programs are well structured and use quite a lot of graphics. It is, however, slightly biased towards the older children.

The reason for my initial confusion is my uncertainty over the book's intended readership: children will gain from the games but what about the parent or teacher who has typed in these rather long programs? They must learn something from the experience! The solution is simple. Why haven't the publishers released cassettes and discs containing the games so the games can be used without the tedium of typing them in.

Despite my minor moan, this appears to be a very good book which offers a good variety of interesting games at a fair price.

Title:	Commodore 64 Color Graphics — a beginners guide.
Author:	Schaffer & Schaffer Applied Research and Development.
Publisher:	Reston Computer Group
Price:	$14.95

THE COMMODORE 64 IS RATHER A paradox. On the one hand it has some excellent graphics modes but on the other you can only use them by resorting to numerous POKEs. Most books give a thumbnail sketch on graphics but leave you no better off. This book is no different. It is based on the creation of tools which enable you to perform certain graphical operations. These include switching in the graphics screen, clearing the screen, drawing dots, lines, boxes, polygons, circles and filling shapes. Clearly such simple material would hardly fill a book so the authors have wrapped it all up in the creation of a sea scene. The idea is that once you know how to draw this scene, you can create your own pictures and amaze everyone.

The book's strong point is the care with which each step of the process is discussed. The overall package is split into sections and presented in small digestable portions. You are invited to type in each section of a program and test run it while you watch how it works. The theory of the operation of that program portion is then described in detail. The descriptions are richly enhanced with diagrams, photographs and sketches. There are also exercises and summaries at the end of each chapter to help reinforce the material learned.

As the book continues, the reader is introduced to more complex concepts. Particular emphasis is placed on the use of shape tables for the drawing of complicated shapes. After high resolution graphics the book progresses on to sprites. Tools for the definition and placing of sprites are provided along with a number of amusing designs. The majority of routines are written in BASIC, but at the end of the book a number of machine code routines are supplied to speed things up.

The actual tools supplied aren't very useful and the work involved in creating a picture is rather excessive, but the book is worth buying for its excellent information on using graphics and its handy ideas.

Title:	Basic is child's play. Commodore 64.
Author:	R.T. Grauer, J. Gordon, M. Schemel
Publisher:	Prentice-Hall.
Price:	£18.30

THIS BASIC TEXT BOOK IS OF AMERICAN origin and is aimed at primary school children. It has been written by two primary school teachers and the mandatory college teacher. The format is typically American in that it makes extensive use of cartoon characters to make the text friendly and exercises to reinforce the points covered.

The book assumes that the user knows nothing about the computer and begins with an explanation of the keyboard and it's functions. Having tapped out a few characters you are eased through arithmetic in direct mode. The book continues at a nice easy pace covering most of the usual aspects of programming including variables, looping, conditionals and arithmetic.

I feel that the book is an adaptation of a general BASIC text with a few tweaks to suit the 64: for example, CHR$(147) is used to clear screens in programs, POKE and PEEK are carefully avoided as are cursor control codes and colours. If, however, you simply want to learn BASIC without any frills, it would do the job.

To conclude, the book is suitable for younger children since it covers a variety of subjects in a readable style.

Title:	Step by Step Programming — Commodore 64
Author:	Phil Cornes
Publisher:	Dorling Kindersley, London
Price:	£5.95

ONCE YOU HAVE BOUGHT A computer you can put it to several uses — games playing, education, business or, better still, programming. A vast range of books aimed at teaching programming is

REFERENCE LIBRARY

currently available. This book offers a different approach: at every stage, there are screen shots of listings and program results.

The book progresses from an initial discussion of hardware to instructions on setting up the machine. There is even a photograph of the computer's interior in case you feel the urge to take your computer to bits. The photographs are clearly labelled so that the function of every key and component is explained.

The remainder of the book gently guides the reader through the rigours of learning BASIC, starting with an introduction to using the keyboard and inputting information in direct mode. Sections on colour, graphics, BASICS, POKE and PEEK, sprites, sound and random numbers then follow. Tables showing POKE and ASCII codes and sprite design grids are at the back of the book. The book may not teach you to write commercial programs, but it will show the basics of what the 64 can do and how to write reasonable programs. Overall the standard of the book is excellent with extensive use of colour and graphics. This book is to be highly recommended.

Title:	Random Alley Adventure
Author:	Michael Orkin
Publisher:	Reston Publishing Company
Price:	£6.95

THIS IS AN AMERICAN COMPUTER book with an interesting and novel approach which should appeal to children in the 9+ age group. The book is unusual in that it deals with chance.

It is set out as an adventure story with a central character called Harold who is a boy with a penchant for computers (he has a desk top and a portable) and adventure. One of his pastimes is going for a bus ride; not to get anywhere in particular but just to have time to think and see new places and meet new people. One day he finds himself in a place called Random Alley. This place is new to him and is populated by some very strange people.

The first person he meets is a compulsive gambler on his way to a roulette game. He is late because at every bus stop he tosses a coin to decide whether he should get off the bus or not. He uses the same tactics to decide which way he goes at cross-roads. Harold, using his pocket computer, tries to show Frank, the gambler, that his coin tossing has an equal probability of showing heads or tails. Harold continues his travels and meets other people each demonstrating a random theory.

Each chapter starts with a short portion of the story describing Harold's adventures. The concepts developed in each story are expanded and illustrated by short programs which the reader can type in. Sufficient detail is given with the programs to show how the program works and to offer some programming guidance. A wide range of ideas are covered including coin tossing, dice, playing card selection and artificial intelligence/random message generation (as in fortune cookies or the program Eliza). The programs are generally short (less than forty lines) so that typing them in isn't too much of a chore.

The story is compelling enough to keep most youngsters interested. The text is augmented with many good quality illustrations, both coloured and monochrome. I found it a most enjoyable and informative book.

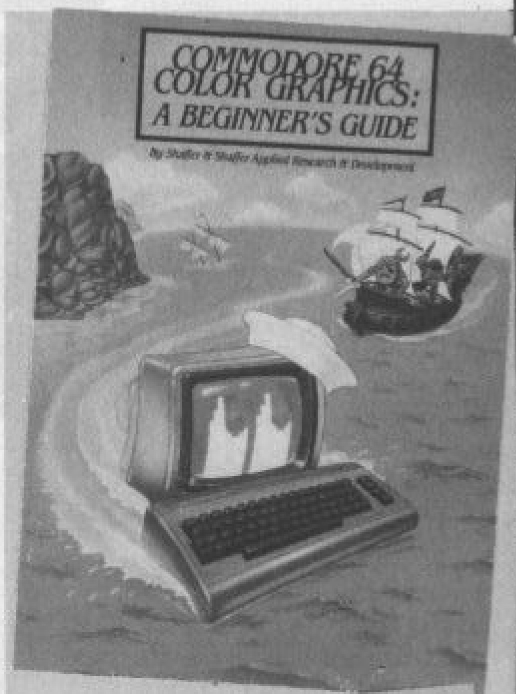
Title:	Intermediate Commodore 64
Author:	Guy Grotke
Publisher:	Reston Publishing Company Inc.
Price:	£14.95

OK SO YOU'VE EXTRACTED ALL THE information from the Commodore 64 handbook and you think you can write a mean program? What do you do next? This book deals with a host of more advanced aspects and is intended to extend your knowledge and expertise. The first two chapters should be compulsory reading for programmers in that they deal with structured programming and the design of algorithms. Using the concepts introduced, it is easy to develop neat, efficient and easily understood programs.

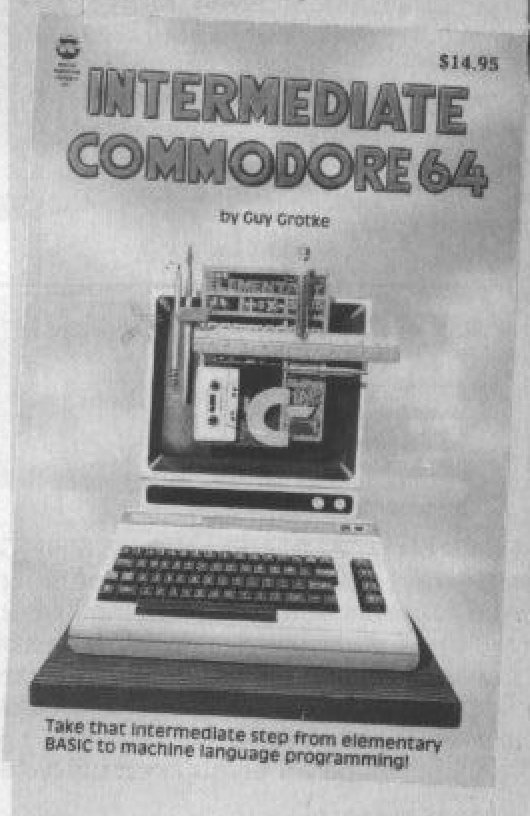
Having established a few good habits, the book slithers smoothly into a tasty secetion on file handling on discs. This starts with sequential files and moves on to random access and ISAM. There is a sadly truncated section on machine code which does little more than discuss assemblers but it does give the author the excuse to use machine code routines in the rest of the book. The remainder of the book covers the use of graphics and sprites. The complex subject of graphics is dealt with well and some useful machine code routines are supplied. A particularly strong chapter discusses the more esoteric aspects of the 64 including the use of the CHARGET wedge to extend BASIC, increasing execution speed and memory banks.

I find the American style a little tiresome but the treatment of quite complex topis is reasonably concise and readable. There is even the odd cartoon to encourage a smile.

Overall this is a very strong book which is a useful source of information and will serve as a good reference guide as well as educational book. In many ways, it is better than the Commodore Programmer's Reference Guide. Well worth a close look.







SOFTWARE SPOTLIGHT

More software steps into the spotlight this month. Our reviewers are here to enlighten you about it.

Toy Bizarre
★ ★ ★ ★ ★
Activision
£9.99
CBM 64 + Joystick

THIS IS WHAT I CALL AN addictive game. It is original and the graphics and sound are superb. The story is about Merton whose job is to maintain the machinery in the toy factory. Simple enough! The problem is that the toys are revolting and to stop them you have to turn the valves off that make the balloons which contain the toys. Now, if that isn't enough, there is this Nora Batty style toy called Hefty Hilda, who turns the valves back on again. So, while pandemonium has broken loose in the toy factory, you have to get rid of the toys. Along the different floors of every screen you will find Piston Platforms. If Merton jumps on a platform when a toy is on the opposite platform, he can stun the toy and get rid of it. He can also get rid of Hilda by the same process.A point to bear in mind is that if you do unto others, then given the chance they will do it to you.

Contained in the game along with the above is a bonus level and a tea break. I won't tell you what happens but I do recommend it if you enjoy frustrating games.

S.L.F.P.



Penguin Study Software: Romeo and Juliet, Twelfth Night
★ ★ ★ ★ ★
Penguin Books Ltd.
£7.95
CBM 64

FROM THE POINT OF VIEW OF operation, use and display, these two programs are essentially identical, and it is therefore convenient to review them together.

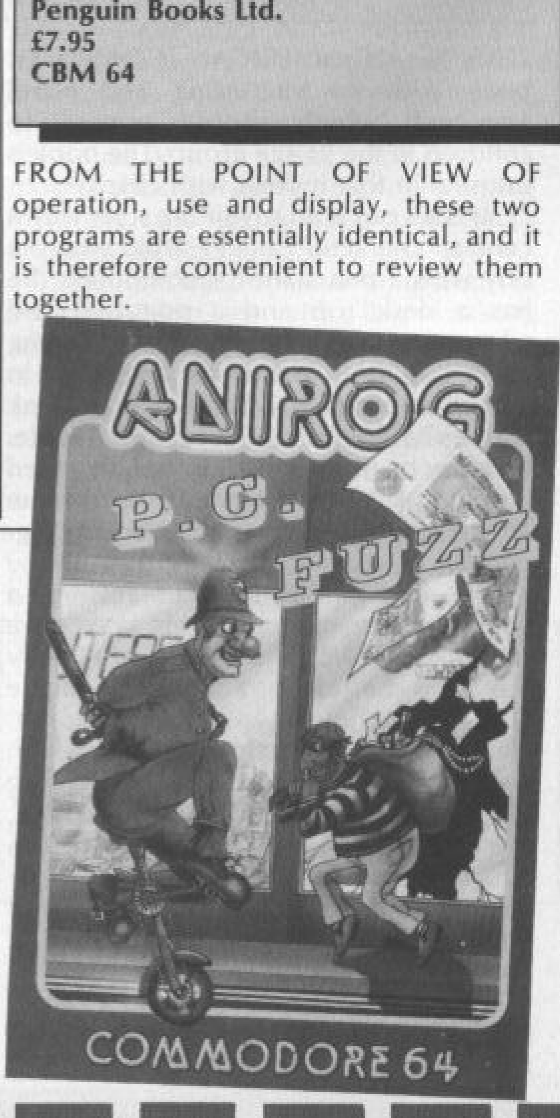
P C Fuzz
★ ★
Anirog
£7.95
CBM 64 + Joystick

IF YOU'RE CONTEMPLATING DIPPING into your pocket for this latest offering from Anirog, then, quite frankly, I wouldn't bother. Is this game slow or is it slow? I realise now why it is called P C Fuzz and not P C Plodd because that would have been giving the game away.

Whilst on patrol on his latest unicycle (can you believe this) P C Fuzz has to prevent the Mob from spiriting away the cash from the banks. The Mob are rolling bombs down the street to knock him of his bike and there is a bunch of punks and drunks out looking for trouble.

You have been given four lives and you lose a life every time you are caught by a bomb or any of the other obstacles. Points are scored for destroying the bombs, and catching the cash-laden balloons with your elasticated truncheon as well as for arresting the members of the mob.

K.M.



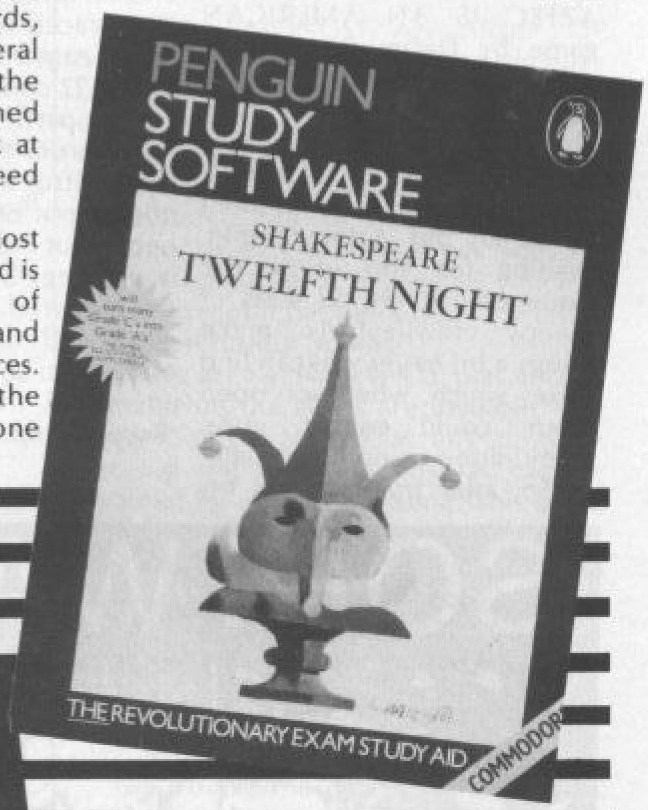
Most of us will remember buying the small potted guides as an aid to passing 'O' level or CSE English Literature. In these packages Penguin have come up with computerised guides which are both easier to use and more informative.

In effect, you are given a dedicated data base containing a huge slab of useful information extracted from the play. The menu gives a list of people and a list of themes which you can incorporate in a search. You can select up to three criteria and the program will search for all incidents or aspects in the play which relate to these criteria. You can choose to search the whole play or selected acts. The results of the search are returned as a series of open-minded comments or questions. Each item is accompanied by suggestions for further searches.

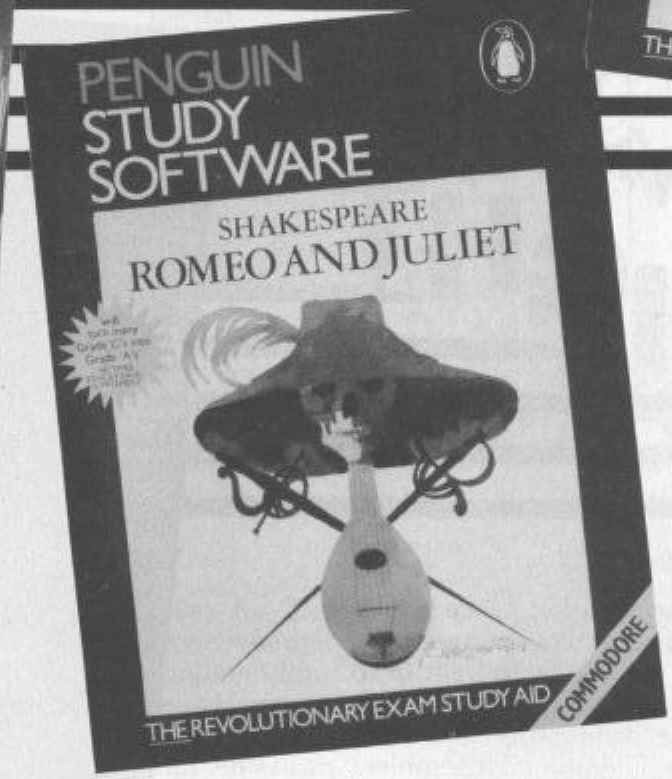
The important point to bear in mind with any educational program is what it will do for you. These programs are not intended to teach you lots of useless facts. For example, they will not tell you who said what, when and why, since most of English Literature questions involve the examination of the themes in the play and the underlying ideas. In other words, these programs will teach you lateral thought and the ability to examine the broader aspects of literature. It is assumed that you have done the spade work at school or college and you simply need your grasp of the play improved.

In operation these programs are most impressive. Loading is by Novaload and is smooth and quick. The selection of criterea is made with the space bar and colour is used to highlight your choices. The search speed is reasonable and the screen presentation is neat. My one complaint is the lack of a hard-copy option. Most users will want to get a copy of the results of their searches to assist study of the text. Overall, two superb packages which are a joy to use and must be worthy of serious consideration.

A.W.







Cave Fighter
★ ★
Bubble-bus
£6.99
CBM 64 + Joystick

I AM NOT TOO SURE WHAT to make of this new game from Bubble-Bus. On the one hand it has good graphics and on the other (for me at least) the game content after a few levels, got repetitive. The game is based on shooting everything that moves and, if it can't be shot at that time, then getting out of its way is advisable. What you have to do is complete as many of the 31 levels of caves as possible. You have a time limit in which to complete each level and very soon it is also a race against the clock.

The graphics used on this game are of a very high standard. The accuracy of the detection is so fine that it becomes a problem when you want to make your man climb a rope for example. To perform this task the man has to be directly under the rope. Technically this is correct but I think it would become frustrating after a while if you couldn't get the hang of it!

As a note to end on, the music was quite good but I do not see any visible connection between Chariots of Fire and this game, except that the man is very athletic.

S.L.F.P.

Jinn Genie
★ ★
Micromega
£8.95
CBM 64 + Joystick

WHISKED OFF ON AN Arabian Knights story that took all day to load (9 minutes, I am exaggerating!). I think it could have benefitted from a fast loader program. Never mind, after reading the game description, it sounded as if it could be a lot of fun. I seem to recall, in the dim and distant past, a game by Synapse called Necromancer. This is by no means the same, but the fighting system used is very similar.

The idea in Jinn Genie is to free the Golden City of the evil caused by a nasty piece of work called Ilfreet. This person has hidden himself away inside the Inner Sactum. Your job, is to get passed his guards, collect a magic jar, free the wise man (imprisoned on the same level as Ilfreet), and capture the Arabian bad guy.

It is essentially a four level game. The first is to land by the Palace. This is where the fighting system is used the most. Whilst flying along you are attacked by the guards. To thwart them you must hit them with your sphere of oblivion (sounds painful!) by moving the joystick while pressing the 'fire' button. To move your carpet just take your finger of the button, and use as normal. The other three screens involve finding an open window, lighting some torches to find the jar and finally, catching Ilfreet himself.

S.L.F.P.

Aztec
★ ★ ★
Beyond
£8.95
CBM 64

AZTEC IS AN AMERICAN game by Datamost and it is produced in the U.K. under licence by Beyond. The main plot behind this Indiana Jones style game is to recover a golden idel which is entombed inside an old Aztec temple. Waiting for you inside the stone edifice are all sorts of creepy crawlies! To make things a bit easier you can find boxes which, when you open them, could contain guns, ammunition or nothing at all.

So, after trying to get my copy to load half a dozen times, I was much relieved to see the text that appeared at the start of the game! During loading it, the screen is adorned by a hi-resolution picture. Once I had chosen my level of play (1 to 8), I was faced with the daunting task of remembering what each of the 22 control keys did!

This particular fact made it a little harder than normal. Why the control movements could not be put on joystick I don't know. But being fair, the game is exciting, once the controls are mastered. The graphics are good and there are a small number of sound effects, but I am afraid the controls were 'Beyond' me.

S.L.F.P.

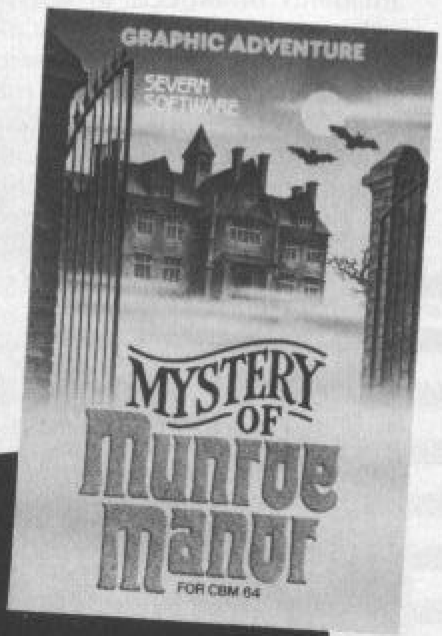
Mystery of Munroe Manor
★ ★ ★
Severn Software
£7.95
CBM 64

AH HA, A GRAPHIC ADVENTURE. A chance to renew old rivalries. It's just like that with me and adventures. I load them. They look at me. I look at them. And it's hate at first sight. I can't explain why, that's just how it is. In the case of Munroe Manor it was one hell of a belter. Old man Bastow has disappeared from Munroe Manor and rumour has it that there is hidden wealth within them there rooms.

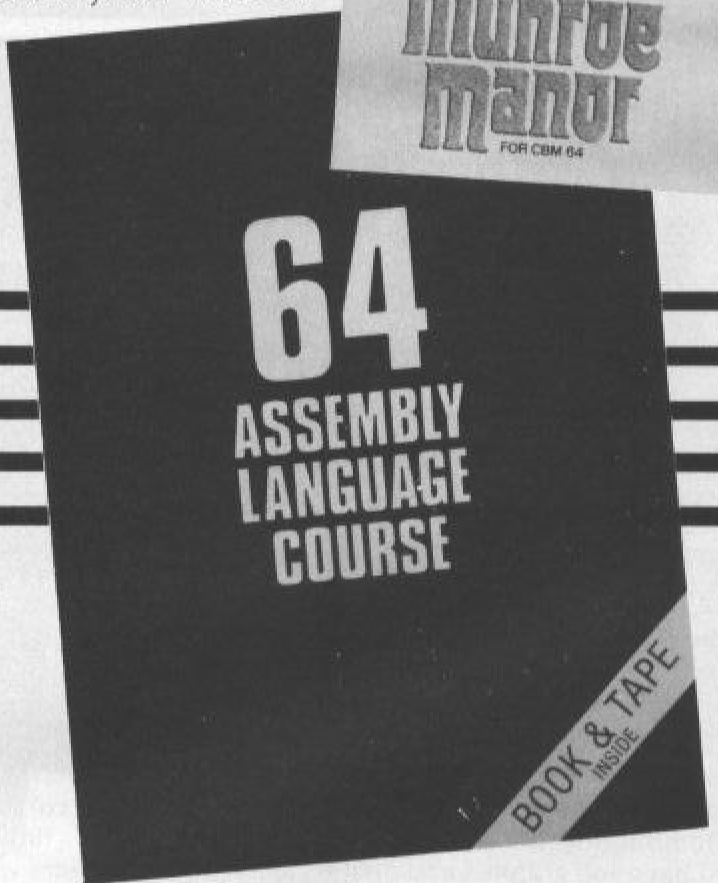
Despite the baying hounds, wandering ghosts and hidden traps perhaps it worth giving it a bash. Solve all the puzzles and mysteries and you could be home and dry. But note that the menu of suggested phrases is very limited and you will have to find your own that work.

Me and Munroe Manor? Well I had my usual run of success in the first room. A death occurred, marked by an epitaph: you have failed after 27 moves completing 0% of the adventure — or words to that effect..Good luck mystery solvers, this looks neat.

K.M.







Beginners Assembly Language Programming for the CBM 64
★ ★ ★ ★ ★
Honeyfold
£12.50
CBM 64

THIS PACKAGE USES THE interesting combination of text and program to ease you through the rigours of learning assembly language. It does an excellent job.

The cassette comes with two utility programs which are rather useful tools. On the 'B' side is a Binary/Hexadecimal Tutor. This program assists in the tedious task of learning the inter-relationships between Decimal, Hexadecimal, Binary and Binary Coded Decimal. The program is menu operated and easy to use. Once you're confident on your conversions, there are exercises built in to test you.

The main program on the tape is a combined assembler/monitor utility. With this program you can assemble machine code routines, disassemble code and run your completed masterpieces. Additionally, you can examine your code and save or load it by use of the machine code monitor included in the package. The monitor has the added advantage that, as normal, it is entered when BRK is encountered. This enables you to protect your program from accidental crashes.

The assembler is fairly comprehensive and includes labels for loops and variables. My main complaint is the use of non-standard mnemonics; it's not really fair to give inaccurate information to beginners, is it? There are insert and move commands to help manipulation of your code and you can convert your final object code to data statements for use in BASIC programs.

The assembler really comes into its own when used in conjunction with the excellent text-book included in the package. This book introduces the concepts of assembly language at a nice steady pace and progresses finally to the use of floating point routines and routines resident in ROM. When you've learnt all there is in the book, it continues to be of value as a reference work giving details of the VIC, 64 and PET ROMs and the full 6502/6510 instruction set.

This is an excellent package which certainly does the job it claims to do.

A.W.



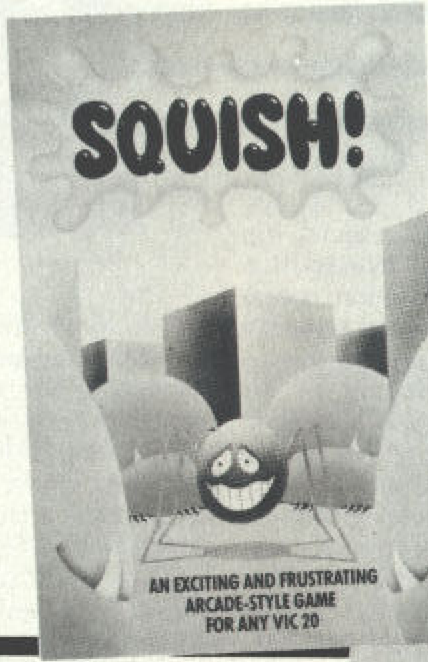
Bricks
★ ★ ★
Palace Software
£5.99
Vic 20 + Joystick

THE BASIC AIM OF THIS GAME IS TO climb your way from the bottom of the screen to the top. In order to do this you have to avoid falling bricks which come hurtling down. Once the bricks settle these are used as your stepping stones to the top. Sounds easy — try it! — More often than not you are hit or trapped — hence you have to start the game again. There are six levels and on reaching level 3 the bricks start disappearing from the bottom of the screen making it even more difficult to reach the top.

The game loaded very easily and is not complex to get started. The graphics are simple — movement jerky and is one space at a time which is very common in the unexpanded VIC 20. The sound is practically non existent apart from the odd Blip — Pong and Explosion.

In its favour the game is quite addictive and I must admit I haven't as yet reached the top which is rather infuriating when my 11 year old brother finds the game a doddle clearing screen after screen. Unlike many VIC 20 games this game can be paused by pressing letter R. Overall quite a good game from Palace Software.

F.W.W.



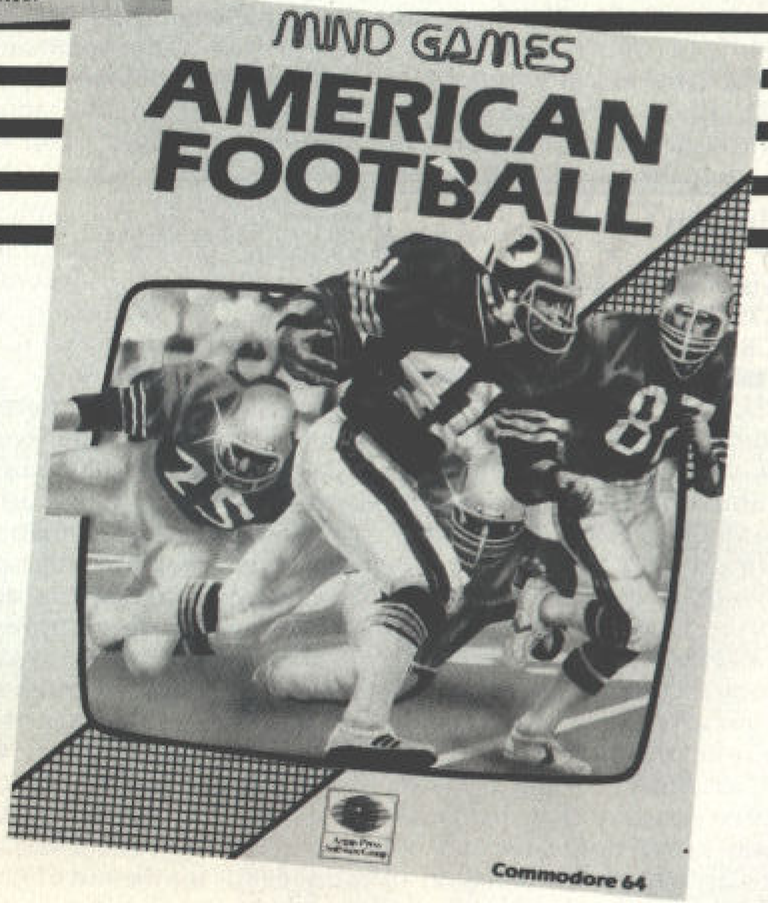
Squish
★ ★ ★
Palace Software
£5.99
VIC 20 + Joystick

TRAPPED AMONGST AN ARMY OF bugs your task is to exterminate the bugs by pushing rocks into them in order to escape. If the bugs manage to make contact, you die immediately. If you are hit by one of their missiles you are megablasted out of existence. If you manage to exterminate them you are warped onto the next level — which is similar except it has more bugs and faster and more furious action.

There are twenty levels of play and on a few levels the aliens are invisible. The game can be paused at any time by pressing the letter R and play can be resumed by using the letter T. This game loads first time and starts immediately. The joystick response is good and the game is not easy, yet not too difficult making it infuriatingly addictive. The graphics are very basic and leave a lot to be desired. The sound is not very good, however these faults do not put you off playing the game.

Not a bad game from Palace software. I would recommend this game to any unexpanded VIC 20 owner.

P.W.W.



American Football
★ ★ ★ ★
Mind Games
£9.95
Commodore 64

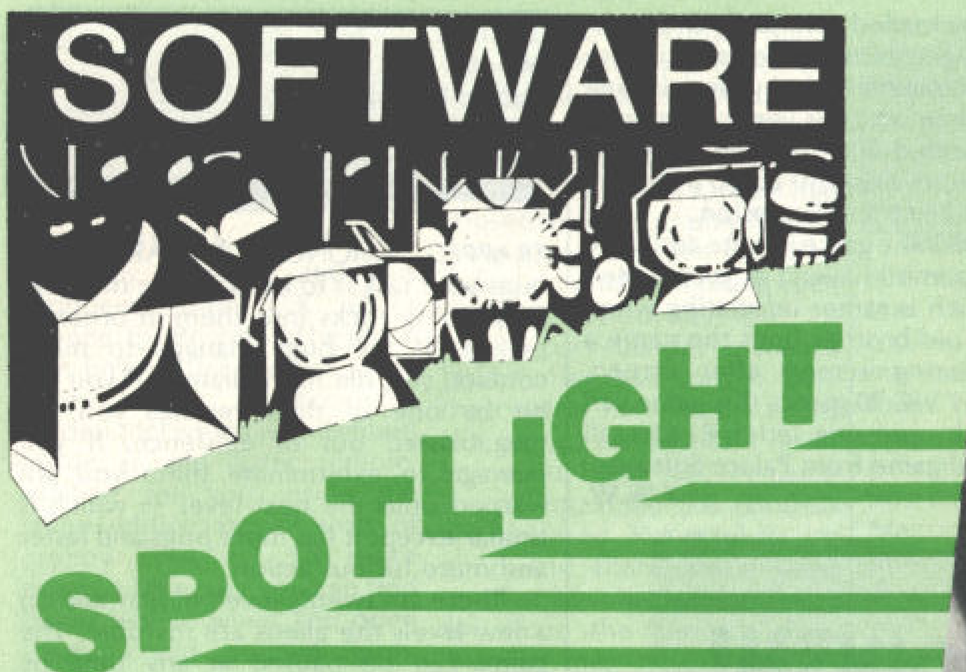
SCREEN, SHOTGUN, POCKET, bomb, draw, sweep, reverse and centre. Sounds offensive doesn't it? Well rest assured it is, but only in the context of football, American style. Yes this is grid iron and these are the offensive passing and running moves in the game. And it's good, real good, particularly if you have a strategic bent.

Each game takes an hour and you can play against somebody else or against the computer itself. But watch out if you choose the latter. It's been programmed to monitor your moves and to try and predict your next one whether you're offending or defending.

Included in the package is a comprehensive booklet explaining all about American Football and how to play the game...computer style. The game period is split down into four quarters. Points are scored for taking the ball into the opponents end zone or kicking a field goal and you do this by outwitting the opposing team with the choice of plays. The plays you decide to make are acted out on the screen by the 'teams'. Other data displayed on the screen includes the yards you have to make and how many plays you have left to do it in before you lose possession of the ball and have to go on the defensive.

Although the graphic representation of the game is not all that brilliant, it doesn't detract from it. The first sixty minutes flashed by and I only lost 24 points to 21 points also proving that you don't need to know anything about it to play. Definitely worth a view I'd say.

K.M.





Basic Adventure Part 1.
★ ★ ★ ★
Honeyfold Software Ltd.
£9.95
CBM 64

IN COMMON WITH A number of Honeyfold Products, this package uses a combination of software and textual information. The idea of this package is to introduce BASIC to the 7 to 11 year olds and as such does a pretty good job.

A fairly novel approach to teaching is adopted by chronically the adventures of Dr Watson and using his activities as the vehicle for teaching. This Wells-like character is whisked into space by his new improved teleport machine and dumped in the space ship belonging to an alien called Aleat and her robot. The hyper-drive of the ship is malfunctioning and the only means of salvation lies in a Commodore 64 in the ship.

From this point, the reader is gently and subtly guided through the concepts of BASIC and the use of the commands. Each step is cunningly slipped into the text so that things progress at an easy pace. The text is nicely illustrated with line drawings and is very well presented. The general idea is that the child can sit at the computer and dabble with the machine whilst reading the book. In effect learning through play.

The tape contains three simple programs on one side which are also discussed in depth in the text. This is a nice touch since it saves a lot of typing. On the other side are four further programs which give further hints and help on the use of PRINT, string manipulation and line numbers. At the rear of the book is a substantial section which gives greater detail on the material covered in the adventure. This is more of a reference section which is for use once the basic material has been absorbed. A comprehensive index in included to provide easy access of information.

A.W.

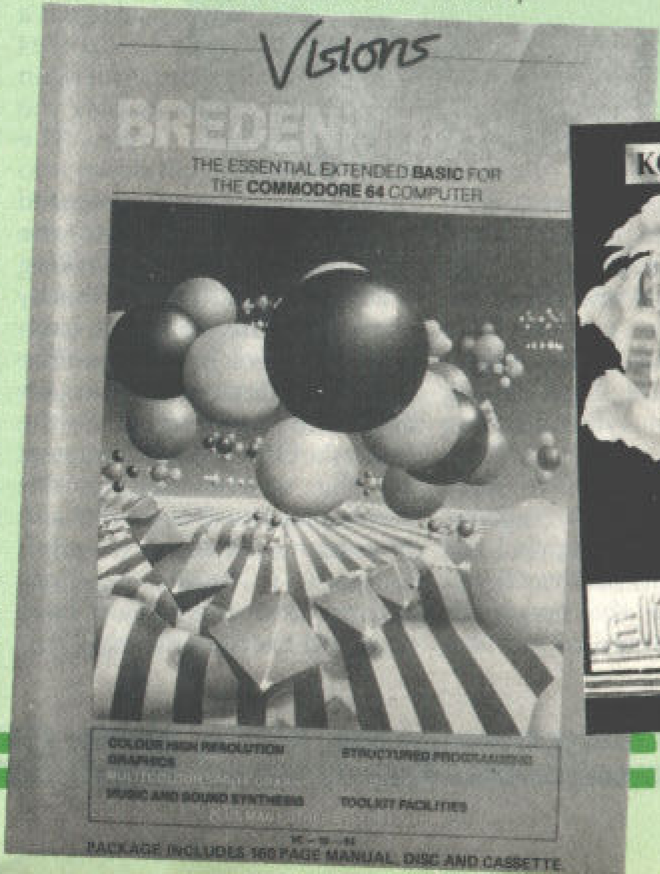
Breden's Basic
★ ★ ★ ★ ★
Visions Software factory
£39.95
CBM 64

BECAUSE OF THE SHORTcomings of the resident BASIC in the 64, a large number of companies are producing extended BASICs of this type. On opening the package you immediately get two surprises. Firstly, there is both a disc and a cassette, giving immediate appeal to all 64 owners. The second revelation is the superb instruction manual. This A5 sized book is approximately three quarters of an inch thick, uses heavy duty card and is a joy to use. A combination of colour coding and summary lists makes access to all commands easy. Use is further enhanced by using each page to describe a single command.

So much for the documentation; how did the software perform? Loading was trouble free and the machine was reset with about 29K of memory left. The manual claimed that some 135 extra commands are supplied, and whilst I haven't counted them, I can believe it. The commands cover the usual areas of high resolution graphics, sprites, sound, structured programming, disc commands, programming aids, error trapping, numeric manipulation and input/output. Most commands performed satisfactorily but were not particularly exceptional. I noted one or two omissions such as RENUMBER, CIRCLE, FILL, and no commands for the use of raster interrupts were offered. Two particular peculiarities were the use of PUT for POKE (to allow for some sort of incompatability) and the use of PERF instead of PROC. Both are certain to cause some confusion.

On the whole, this is a well written package which operates without any obvious bugs. The diskette contains a number of demonstration routines which show the capabilities of the system well and should provide guidance on the usage of commands. At the price, this extended BASIC represents excellent value for money.

A.W.





Strike Force

★ ★ ★

Bubble Bus

£6.99

CBM 64 + Joystick

SHIFT, RUN/STOP, LOAD and prepare to meet thy doom. The Strike Force is on its way and you are the sole defender against what is undoubtedly a very strong force. But just because you are outnumbered and the title screen serenades you with that catchy little number Suicide is Painless, don't be down hearted.

The object of the game is to shoot and destroy each wave of enemy attackers from the first Helicopter wave through to the bombers and the armoured personnel carriers and supply trucks. You have a set time period in which to destroy each enemy wave and 5 lives before you yourself are killed. Your defences consist of a central set of on screen gun barrels which can be raised or lowered in terms of their range by pushing forward and pulling back on the joystick.

Beginning to despair? Well don't. An edit facility has been incorporated to help you stack the odds more in your favour. This decreases the size of the enemy force on a difficulty level of 1 to 5, slows the enemy down, lessens the accuracy you need to destroy the enemy and increases the rate at which you can fire your guns.

K.M.



Tazz

★ ★ ★

Bubble Bus

£6.99

CBM 64 + Joystick

AAGGHH! THIS IS ONE OF those screamingly frustrating arcade games in which you can see the enemy about to strike but you haven't got the coordination to get out of the way. And then, having lost, it immediately gets you coming back for more in a vain attempt to beat your previous high score. Yes, that's Tazz all right.

You are in charge of a little man who can move and fire up, down, left and right only he can't do both at the same time. The object is to clear 10 rooms and then 10 'super' rooms of between 20 and 230 stationary bugs and a variety of marauding nasties. Each room has a different theme from masks and evil eyes to butterflies and windmills. You have to direct the yellow peril to clear the room before the advancing walls crush him. After every two rooms there is a bonus room, but be quick to collect the bombs as this is the only way to destroy the windmills later on. Using the bombs will also provide a temporary lull in the marauding nasties and advancing walls. Any key detonates the bombs.

You start off the game with a generous 5 lives and you'll need them if you are able to reach the cocktail room at the end of the road. An extra life is gained for every 10,000 points gained and a life is lost if you run into a bug, touch a nasty or get crushed by the walls. Make no bones about it, this is a good, solid arcade zapping game.

What it lacks in innovation it more that makes up for in frustration value.

K.M.

Zenji

★ ★ ★ ★

Activision

£9.99

CBM 64 + Joystick

NOW HERE'S A LITTLE BRAIN-teaser for you. I do not know how much of the words written on the back are true, but according to Rokan, a master of Zenji, you have to connect all the elements together by their paths and when this happens, Zenji occurs. So, that's the theory behind the game and I can say that I enjoyed it immensly.

When the program had loaded, I was greeted by an imitation of Chinese music. Pressing 'F1', got me onto the first screen in which a small set of paths are unconnected. By traversing the connected paths, and rotating them, you can connect up with the unconnected paths. As soon as they are all joined, the screen is finished, your points are totalled up and your character is moved onto level 2. On each progressive screen, more paths are to be made.

It sounds easy, but when there are time limits on each screen I found it presented more of a challenge. Another bonus is that on the first few levels a segment starts to countdown. If you reach it before it gets to 0, extra points are awarded. If you do not reach it in time, it disappears. I found that occasionally, this stopped me from finishing the level. It's worth seeing it, if possible, but then, you don't have to take my word for it, take Rokan's.

S.L.F.P.

Kokotoni Wilf

★ ★

Elite

£6.95

CBM 64 + Joystick

WHAT'S ALL THIS GUFF? A HYPER-active imagination at work on the sleeve notes. Let me summarise. There was this magician geezer what learned of a valuable amulet that had been smashed to pieces and scattered through time. Too long in the tooth to go and fetch them himself, he enroles this Kokotoni Wilf character to do the leg work and even slaps a pair of wings on his back to make him more mobile. As the controller of the joystick you have become Wilf's guiding mentor. With a string of spare lives tucked under your belt, you are urged to travel back the 500 million years in time to gather the first fragments, find the glowing time gate to another period and gather yet more fragments. Goody, goody.

At the end of it all there's a message from the magician telling Wilf why he was sent on the godforsaken quest. Be one of the first to tell Elite all about it and you could be one of a lucky 100 to receive a free copy of its next game, The Fall Guy. Needless to say this is little more than a maze game on a scrolling screen and a central character not too responsive to careful direction. Not exactly much to write home about is it?

K.M.





The BASIC Facts

Improve your BASIC programming with advice from A.P. and D.J. Stephenson on strings, randoms and time.

BASIC PROVIDES A SET OF keywords which can perform complex surgical operations on strings. Separate characters, or blocks of them, within a string variable can be discarded, rearranged, reversed in order or changed into equivalent numerical or other forms. Before discussing the function and syntax of the keywords it is pertinent to enquire why we would ever want to mess around with letters within a word. Take your name and initials as a starting point. If, at some stage in a program the screen message "Enter your name" appears, you would probably answer in the normal way — for example: Jimmy Bloggs or J Bloggs. Within the computer, it would be more convenient, both for sorting and searching purposes, if the surname was stored first and the initials or forename/s last. This would entail using one of the suitable string handling keywords for rearranging the input obtained from the keyboard.

String handling is also useful for reducing the amount of input required from the keyboard. For example, a program which tests knowledge in the form of questions and answers on capital cities of countries should not penalise spelling errors. To avoid rejection of an answer on spelling grounds a program can be arranged to accept, say, the first three letters of a city — TUN could be accepted for TUNISIA. It is convenient, in fact almost essential, for individual records within a file to be accessed by the full name or by the first one or two characters of the name. It is even possible to arrange for a few characters in the middle of the full name to be sufficient for locating it from within the file. This is not only to save time during keyboard entry but because the full name may have been forgotten, in which case the record would be 'lost' forever. Converting plain language messages into coded form and subsequent code cracking is another area which can benefit from subtle use of string handling keywords. There are scores of other uses but it is time we started examining the *keywords* and examples. Several short programs, or rather bits of programs we shall call *modules*, will follow. They are described in the text rather than by the usual REMarks. When you are learning programming dodges, REMs often make modules look frightfully complicated — the opposite effect to that intended. After you learn these dodges and start writing lengthy programs of your own then REMarks are of great value.

Finding length of a string

A program will frequently want information on the length of a string. For example, how many characters there are in, say, A$. The keyword to use is LEN, the format being:

```
L = LEN(A$)
```

Example:

```
100 INPUT"ENTER A WORD OR SENTENCE ":A$
110 L = LEN(A$)
120 PRINT"YOU HAVE ENTERED "L" CHARACTERS.
130 GOTO 100
```

This is, of course, an *endless loop* so you can only get out of it by pressing the RUN/STOP key. Try it with dozens of different inputs to confirm that all characters, letters, numbers, punctuation and even spaces are included in the total length. Note carefully that although A$ is a *string* variable, its length L is a *numeric* variable.

Splitting off leftmost characters

The key word here is LEFT$, and the syntax is:

```
LEFT$(variable,N)
```

where N is the number of characters to be split off from the left. For example, if A$ contains the word ARGUS, and we write B$=LEFT$(A$,3), then B$ will contain the letters "ARG". Clearly, we can't split off more letters than the original word. The following few lines can be used to try out LEFT$ with different inputs:

```
100 INPUT"ENTER A WORD ";A$
100 INPUT "HOW MANY CHARACTERS TO BE SPLIT OFF ";N
120 B$=LEFT$(A$,N)
130 PRINT A$
140 PRINT B$
150 GOTO 100
```

Note that the original word in A$ is *preserved* intact and not corrupted in any way by the splitting.

Splitting rightmost characters

The keyword here is RIGHT$ and the syntax is:

```
RIGHT$(variable,N)
```



This can be tried out using the same test program as above but with line 120 changed to RIGHT$(A$,N).

Let's try something a little more ambitious; assume 100 assorted words are in an array A$(N) and we want to print out only those words having four letters in which the first letter is F. It could be done as follows:

```
100 FORN=1 TO 100
110 IF LEN(A$(N) )=4 AND LEFT$(T$(N) )="F" THEN PRINT
A$(N)
120 NEXT
```

Make sure that some earlier program segment has indeed placed the assorted words in the array A$(N) — it won't work without it! Note that the AND connective in the IF statement ensures that only those words which satisfy *both* conditions are printed out.

Sometimes, we may wish to split off one or more characters from within, rather than from one end of, a string. The relevent keyword is MID$ and the syntax is:

```
MID$(variable),S,N)
```

N is the number of characters to be split off, starting S characters from the beginning. For example, suppose the string variable A$ contains "CUCUMBER" and we write, B$=MID$(A $,3,5). This will split off 5 characters, starting from the third character. That is to say, B$ will contain "CUMBE". MID$ is an awkward and often misused BASIC keyword so a few extra examples are justified. Assume that A$ contains "EXCOCET":

```
If we write B$MID$(A$,3,3) then B$ will contain "COC"
If we write B$=(A$,2,5) then B$ will contain "XCOCE"
If we write B$=MID$(A$,5,3) then B$ will contain "CET"
If we write B$=MID$(A$,1,1) then B$ will contain "E"
```

You may have noticed that in the last example we could have used LEFT$(A$,1) instead of the MID$ form. Situations such as this often arise in which there is a certain crossover in function between keywords.

Let's see if we can enter a word or sentence at the keyboard and print it out backwards — it may be of some use in a word game:

```
100 INPUT"ENTER A WORD OR SENTENCE ";A$
110 L = LEN(A$)
120 FOR N = L TO 1 STEP -1
130 PRINT MID$(A$,N,1);
140 NEXT
```

Since the FOR loop is stepping backwards, the first character printed is the last character of the original. The semicolon at the end of line 130 ensures the printout is all on the *same line* — providing, of course, there is room on the one line. If you miss out the semicolon, the printout will be in the vertical plane.

Converting a number to a character

Computers can only store numbers. The ability to 'store' letters, punctuation marks and various other characters is an illusion. The fact is that all characters available on the keyboard are given a specific code number. The code in general use since the early days of computing is the American Standard Code for Information Interchange, mercifully abbreviated to ASCII. The full code is given in the User Manual and is repeated ad nauseum in nearly every computer book published so only a few relevant entries in abbreviated form are repeated below:

ASCII CODE

Digits	Upper case letters
0 = 48	A = 65
1 = 49	B = 66
..	..
9 = 57	Z = 90

The keyword for converting the ASCII number to the corresponding character is CHR$, using the syntax,

```
CHR$(N)
```

For example, if we enter the line:

PRINT CHR$(66),

the screen will display the character "B". The following few lines will show the correspondence between ASCII numbers and the upper case letters:

```
100 FOR N = 65 TO 90
110 PRINT CHR$(N) " ";
120 NEXT
```

The ASCII codes, 0 to 31 have not been standardised and are left for Control Codes, specific for a given computer. In fact the Commodore 64, pays only lip service to much of the standard ASCII code because of the need to squeeze in a range of special graphic symbols. However, the digits, upper case letters and standard punctuation marks remain as pure ASCII.

Converting a character to code form

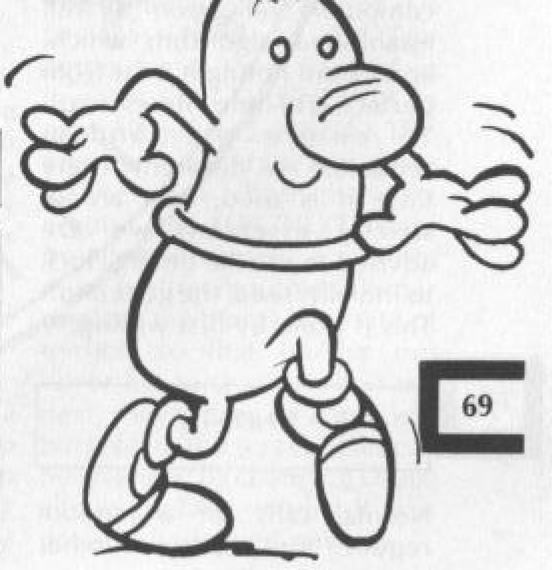
As we have seen, CHR$ converts a code number to the corresponding character. For the reverse process, that is to say, converting a character to its equivalent code, the key word ASC is used, the syntax being:

```
ASC("character")
```

For example, if we enter PRINT ASC("A"), the number 65 is displayed. Note that the literal character must be enclosed in quotes. You are not obliged to consult the User Guide every time you need the code number for a character. Simply enter the above PRINT line in *direct mode* and the code, including any of the special graphic characters, will be displayed, subject to the proviso that the character is displayable. (Control codes do not have printable characters.) The character can be in string variable form instead of literal. For example, PRINT ASC(D$) will display the code for the character stored in D$. However, if D$ contains more than one character, only the code for the *first* character is displayed.

Numeric and string conversions

It is often required to convert a numerical value to its string form or vice versa. As we discussed in an earlier part of the series, string variables and numeric variables cannot be mixed. We cannot write, A=B$ without invoking a nasty error message from the BASIC interpreter. However, it is possible to restore harmony, in spite of the mismatch, by using A = VAL(B$). The two conversion keywords are VAL and STR$, the syntax being:



```
VAL(string variable)
```

and

```
STR$(numeric variable)
```

Example: A$ = STR$ (N) will convert the number in N to string form in A$. If N is 457, then A$ will contain "457".
Example: N = VAL(A$) will convert the string form number in A$ to the pure numeric form in N. If A$ contains "457", N will contain 457.
It is sometimes convenient to store numbers in an array in string form, using STR$ for the conversion. However, when the numbers in string form are eventually extracted from the array it will be necessary to convert them back to numeric form with the aid of VAL. You will remember that normal arithmetic can not be performed on numbers unless they are in numeric form. VAL will return a value of zero if the first character in the string is non numeric. For example, if A$ contains "D235" and we write N = VAL(A$), then N = 0 because subsequent characters after the first are ignored.

Random numbers

A pseudo random number can be obtained with the aid of the keyword RND, using the syntax:

```
RND(number)
```

The random number generator in the Commodore, and indeed in most other makes of computer, relies on a well established algorithm which, apart from noting it is far from perfect, is of little interest to us. To ensure the random sequence is different every time it is used, (not always strictly essential) we are advised to use the timing clock to initially 'seed' the generator. This is done by first writing,

```
R = RND(-TI)
```

Normal calls for a random require *any positive number* within the brackets. It doesn't matter what number so we might just as well use "1" every time. The number we get in R by using R = RND(1) is always a multi-digit fraction within the range 0 to 1 although it is never 0 and never 1. However, such a number is seldom practical so in most cases, a certain amount of jiggery pokery is needed in order to obtain values within the desired range.

1. To obtain a positive random integer between I and N, use:

```
R = INT(N)★RND(1) )+1
```

2. To obtain a positive random integer within a range X to Y inclusive, use:

```
R = INT( (Y+1–X)★RND(1) )+X
```

where Y is the higher limit and X is the lower. Note that Y must have an extra 1 added.

For example, to obtain a random number between 1 and 6, such as the throwing of a dice use

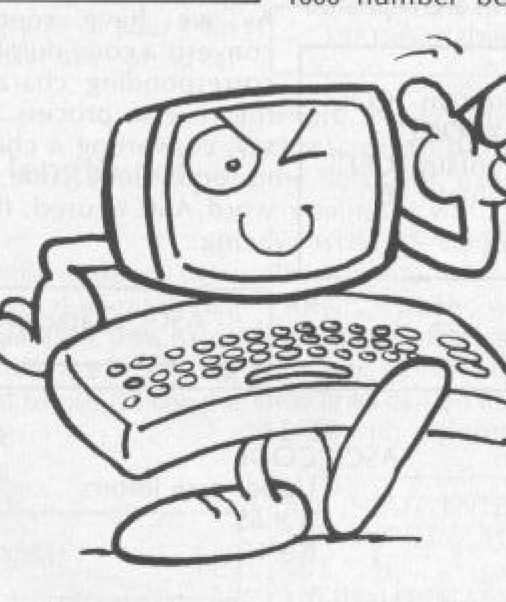
```
R = INT( (6)★RND(1) )+1
```

To obtain a random number between 65 and 90 (note this will is the ASCII range for upper case letters) use

```
R =INT( (91-65)★RND(1) )+65
```

This, of course, simplifies to

```
R = INT(26★RND(1) )+65
```



The following few lines are useful for trying out the effect of RND with various limits:

```
100 INPUT"ENTER HIGHEST LIMIT ";Y
110 INPUT"ENTER LOWEST LIMIT ";X
120 FOR N = 1 TO 20
130 PRINT INT( (Y+1–X)★RND( 1) )+X
140 NEXT
```

If you have a suspicious nature, you may want to test the randomness of RND. When a set of random numbers are displayed, it can often appear that certain numbers seem to be thrown up too often. This effect is particularly noticeable if the range of numbers is small.

Even pure random numbers can play tricks sometimes, though legitimate tricks. For example, a perfect random number generator could output 10 sevens in a row. A random sequence does not mean all the numbers must be different. The probability of any particular sequence is just as likely as any other so 10 consecutive sevens may be surprising but not necessarily a cause for suspicion unless the same thing happens every time. However, there are several ways in which you can test the 'randomness' of RND. The easiest way to generate a *large number* of randoms and then find the average. In other words, add them up and divide by the number of numbers to find the average. The average should be close to half the range. For example, generate 1000 number between 1 and 1000. If the average is close to 500, it indicates there is little bias to one side so the generator is good. The following short module will scroll 1000 randoms on the screen and end with the average random:



```
100 T=0
110 FOR N = 1 TO 1000
120 R = INT(100★RND(1) )+1
130  T  =  T+R:AV  = T/1000
140 PRINT R" ";
150 NEXT
160 PRINT"-----------------------
----------"
170  PRINT"THE AVERAGE RANDOM IS "AV
```

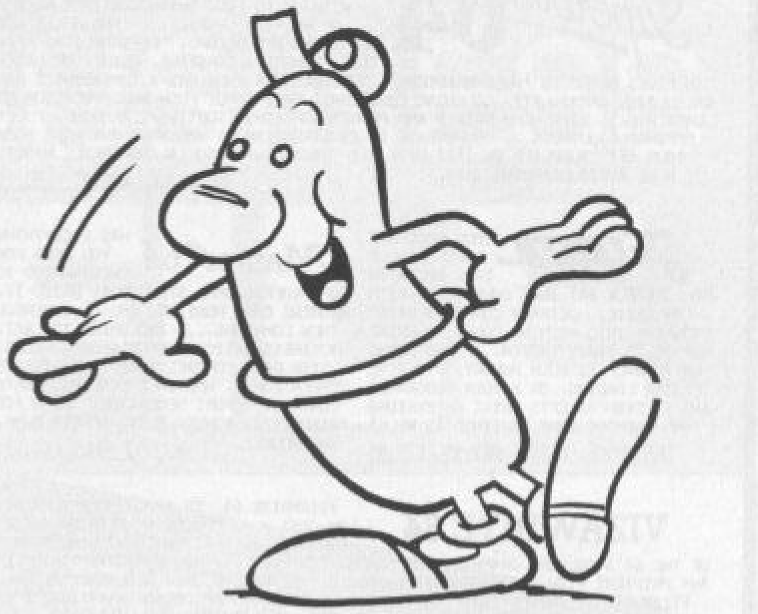


The average should be close to 500 each time it is run. Sometimes it will be, say, 499.4 and other times, 501.6. The actual figures are unimportant providing they are reasonable close to 500 and not consistently and heavily biased to one side.

Random test words

To consolidate some of the previous work, study the listing below. It is a program module for generating a set of random characters. It can be spliced into any program which is designed to try out string manipulations.

```
100 PRINT CHR$(147):Y=90:X=65:REM ASC RANGE FOR
LETTERS
110 FOR A = 1 TO 10
120 FOR R = 1 TO 10
130 N = INT(Y+1–X)★RND(1)+65
140 A$(A) = A$(A)+CHR$(N)
152 NEXT
155 PRINT A$(A)
157 NEXT
```





The module, as it stands, will generate and load the array A$(A) with 10 random "words" each of 10 characters. The outer loop is responsible for ensuring there will be 10 numbers and the inner loop generates the 10 random letters which form each word. The words are, nonsense but this doesn't matter — the more random, the better it performs as a test pattern. Note that line 150 concatenates each random character to form a 10 letter word. The characters are obtained by using CHR$ to convert the random number to an upper case letter. The line numbers are ragged, a sure indication that we developed it by trial and error — an admission of human frailty for which no apologies are offered.

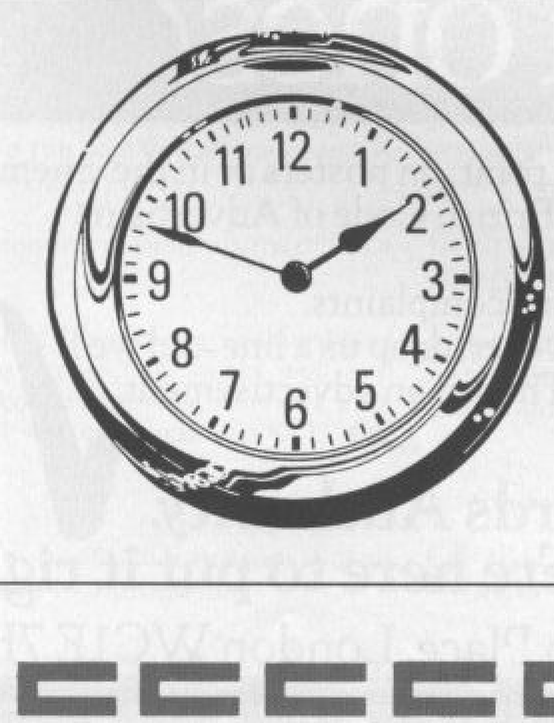
Time and time delays

All the complex functions within a computer are controlled by a central oscillator known as the master oscillator or 'clock'. The frequency is very high — millions of pulses per second in order to create the illusion (sometimes?) that your programs produce results 'instantaneously'. To make use of this oscillator for normal time keeping, a special silicon chip is used for *counting-down* the high frequency to a reasonable value. The basic unit is called the 'jiffy' which is 1/60 of a second. As soon as the computer is switched on, the clock starts and the jiffies go up and up. There are two special variables, T1 and T1$ which the computer keeps continuously updated so they contain the current elapsed time.

T1$ contains a six digit number in the form hours, (first two digits), minutes, (middle two digits), and seconds (last two digits). For example, if 3 hours, 45 minutes and 7 seconds have elapsed, TI$ would contain the string "034507". It is possible to set TI$ to any time, including the 'real' time by a simple assignment. To set the clock to quarter past five exactly, write TI$ = "051500". The other variable, TI, is pure numeric in form, recording elapsed time in total jiffies. You cannot directly alter the value of TI although it is automatically reset when you assign a new value for TI$. To watch the clock go up and up, try this.

```
100 T$ = "000000"
110 FOR N = 1 TO 10000
120 PRINT TI$
130 NEXT
```





Time delays

Occasions arise when it is necessary to slow down a program by introducing a time delay. The lazy way is to use a FOR/NEXT loop which simply counts up from 1 to N but does nothing more. For example,

```
100 FOR N = 1 TO 1000:NEXT
```

If this line is interposed within a program, everything is held up while the computer counts up to 1000 — which won't be very long. The more elegant solution is to make use of TI. a delay of 5 seconds can be introduced by the lines,

```
100 D=TI
110 IF TI < D+300 THEN 110
```

The instantaneous value of TI is first assigned to D. The computer is then trapped in line 100 until TI climbs up an extra 300 jiffies (5 seconds). This method is accurate and more convenient than messing around with FOR/NEXT loops. In games programs, a random delay is often required in order to place the player in a state of tension so that he/she can never be quite sure when the next megapowered laser will burst on to the screen. This can be arranged by changing D+300 above to D+R, where R is a previously derived random.

BUSINESS FILE

This month, David Crisp gets down to business with Vizastar, Help and Superhelp.

HELP/SUPERHELP
STACK 100
£15.00/£25.00
Commodore 64

HELP AND SUPERHELP ARE A PAIR OF programmers aid cartridges. They are both basically the same unit except that the SUPERHELP cartridge contains a few extra commands as well as a two pass assembler. Stack 100 split the cartridges up into four parts in the case of Superhelp and three parts for Help.

Part 1

Part 1 contains additional BASIC commands.

HELP. This command will display the last line carried out after an interruption. In some cases it is not very reliable.

FIND. This will allow you to get a list of all the lines in which you have used a particular string variable or command word. I often use this when I have used variable name twice.

DELETE. This is a line Delete using standard TO-FROM syntax. It works reliably and is very important if you renumber set subroutines.

GENLINE. Most people know this function as AUTO. It produces line numbers automatically when writing a program. You can indicate the starting line number and the increment. I find this a useful time saver.

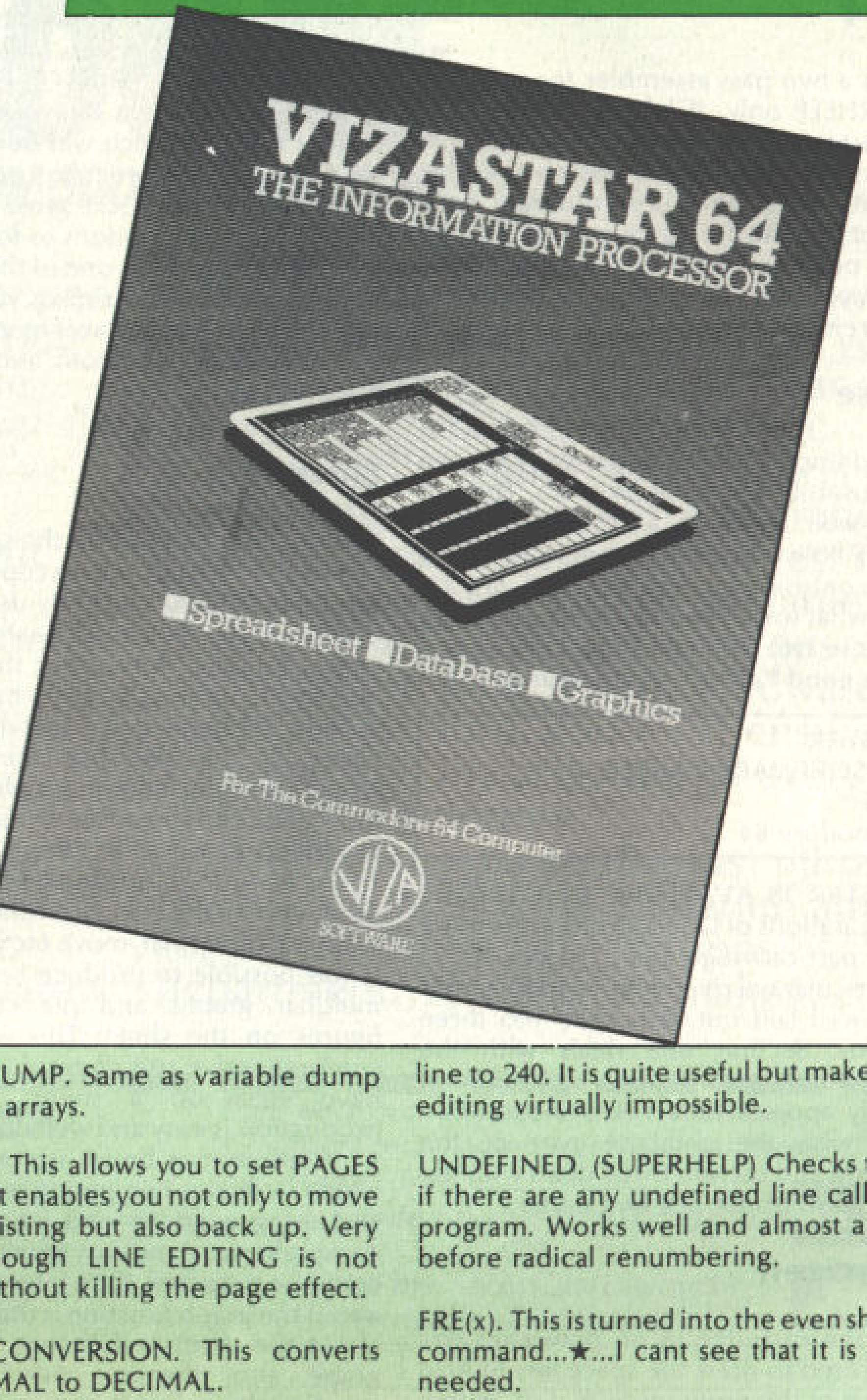
TRACE. Almost vital for those of us who write slightly less than perfect programs. This enables you to STEP through a program line by line in order to see where things went wrong.

SINGLE STEP. As TRACE but the program only goes to the next line after a keypress a sort of IN DEPTH trace.

END. This command simply turns of trace and step.

RENUMBER. One of the few renumbers I have come across that works well and reliably. I could not make it misfire and, best of all, it copes with GOTOS and GOSUBS.

VARIABLE DUMP. Particularly useful. DUMP the current values and names of all variables used in a program.



MATRIX DUMP. Same as variable dump but dumps arrays.

PAGE LIST. This allows you to set PAGES for listing. It enables you not only to move DOWN a listing but also back up. Very useful although LINE EDITING is not possible without killing the page effect.

HEX-DE' CONVERSION. This converts HEXIDECIMAL to DECIMAL.

FILE OPENER. The '(' key enables you to open a numbered output file without the usual rigmarole.

KILL. This kill command SOFT-SWITCHES off the cartridge.

COMPACT. (SUPERHELP) This command increases the possible density of a single line to 240. It is quite useful but makes line editing virtually impossible.

UNDEFINED. (SUPERHELP) Checks to see if there are any undefined line calls in a program. Works well and almost a must before radical renumbering.

FRE(x). This is turned into the even shorter command...★...I cant see that it is really needed.

Part 2

This part is a disassembler/monitor. The disassembler is very good and easy to use. It is easy to modify the contents of addresses in hex and the display is so clear that I have used it as a teaching aid. It can be
have used it as a teaching aid. It can be

used to move blocks of data around in memory but the documentation for this isn't very clear. The monitor is not in the Commodore style and some of its uses are not very obvious.

Part 3

Part 3 is a DOS SUPPORT SECTION. It gives the 64 some of the disc handling commands that Commodore left out thus making it easier to get a directory of the disc, open and close files, read channels, save, load and verify.

Part 4

This is a two pass assembler for use with SUPERHELP only. If I had seen this as a novice to assembly language I would have run a mile. It is a two pass assembler and as such it stores the source program onto disc for reading during the second pass, so it can only be used with a disc drive. The documentation is not very clear or comprehensive.

In use

The additional BASIC commands are the most useful part of the program. They all work well although a couple of them seem a little pointless. The monitor was a little confusing at first and will not be quite what some people expect. But, all in all, these two cartridges are a good buy with a good balance of commands.

VIZASTAR
VIZA SOFTWARE LIMITED
£99.95
Commodore 64

VIZASTAR IS AVAILABLE IN VARIOUS configurations of Disc/Cartridge etc mine being part cartridge and part disc.

The manual is thick, readable and fairly well laid out. It is split into three sections. Section one deals with the spreadsheet part of Vizastar as that is what initially appears on screen. Section two deals with the database part of the program and section three deals with the integration of the first two parts.

On screen

With the cartridge in on power-up you are invited to press the space key to load the rest of the program from disc. Within seconds an intro screen appears. Loading is reasonably fast as a lot of the GEN is held on the disc.

The initial colours are very good irrespective of whether you are using a colour screen or a green screen monitor and, with the inbuilt colours, the information is easy to read. You also have the option to change colours.

When the software has loaded you are presented with a special spreadsheet. Lines up and down the screen make 'zero'ing' in on a particular cell very easy, and whereas with most spreadsheets the active cell is highlighted by being inverse, in the case of Vizastar a small arrow points to the relevant cell. I have never seen a spreadsheet display quite the same as this before on the 64 but I loved it. It had a very efficient and professional feel to it.

Initially the screen remains fairly uncluttered, with little information about options displayed. But with one press of the CBM key, a large panel saying SELECT appears superimposed on the spreadsheet. A broad menu is displayed on the topline this covers options such as CELL, SHEET, FILE, PRINTER, GRAPH and DATE. Below this is a sub-menu showing further options which will be available if you choose a particular option. (For example choosing CELL gives you a sub-menu containing options to format erase move etc). Choosing one of these further options in many cases gives you another sub-menu. This multi layer menu system is easy to use, very smooth and enhances the program.

Entered in

I felt that the best test for the spreadsheet part of the program was to copy data from the spreadsheet I currently use onto this one. I entered data onto the sheet quickly and accurately. Because of the well laid out menu options I rarely had to refer back to the manual. It is not necessary to cover all the available commands. It would be confusing for people who have not used a spreadsheet before and repetitive reading for those who have. There are no glaring omissions on this sheet and all the standard functions such as replicate, format, move etc were there. It was possible to produce line, bar and multibar graphs and pie charts from figures on the sheet. This is the most comprehensive use of graphic facilities I have seen on a sheet. Bar chart production is easy and well documented.

Windows can be set up on the sheet which facilitates comparing sets of figures and using windows and graphs. A very comprehensive and easy to read summary could be prepared. One interesting thing about the graph function is that you move about the sheet and change figures the graph also changes. Admittedly this function does slow things up a little.

Hardcopy

Using the options it was possible to find a way of obtaining a printout with most printers. If your printer is capable it is possible to get a screen dump of graphs etc. Normal routines are used for getting printouts of figures and data and through the printout menu it is possible to indicate whether you have Commodore/non-Commodore printer, single sheet or tractor fed paper etc. It is not possible to give any definite list of which printers work but I tried it with a SMITH CORONA TPI, a Shinwa CP80 and a COMMODORE MPS801. All performed well.

Even more

As a spreadsheet this is excellent. As a combined spreadsheet and database it is positively amazing. Using special commands available from the menu it is possible to get from the spreadsheet into the database. There is full integration between the two and the potential to program the whole set-up makes the writing of applications possible. Basically if you have an invoice and statements routine set up on the database it is easy to store the information on the spreadsheet. If you have used a database management system before you will have half an idea of what this integration is about. Writing a program to produce a running application through is rather different and requires a fair amount of re-learning. But it's well worth the trouble.

Auto demo

The demo routine on the disc makes everything far clearer than I ever could in a review. It runs automatically and shows how much integration there can be between the database and the spreadsheet. There is a demonstration invoice routine which could be padded out to produce a fully blown application but it's main use is as a tutorial. The main commands, and how to use them, are explained in the demo. At first glance it seems mind boggling but, after working through the examples a couple of times, becomes much clearer. Quite a lot of hard work and concentration is needed to program the database.

Worth the effort

Although it will take a long time to become proficient at using this to its full potential I am determined to persevere with it. If you use a spreadsheet and a database or if you think you are likely to in the future then bear this one in mind. As a stand alone spreadsheet it would be quite expensive but even if that was all it did I would still consider buying it. As a combined package I think it is very reasonably priced. As a stand alone database it does not appear to be as powerful as some of the dedicated packages, but if you require a database which is biased towards storage and use of numeric data this would be a very good choice.

COMPUTERS IN BUSINESS

Alison Hjul has braved the streets of Sheen and visited the Micro Computer Centre and its staff.



IN PURSUIT OF A TARGET for my exposé of a high street Commodore dealer, I had to look no further than my local Micro Computer Centre. I sympathised with Mr. Grant and his staff for, like myself, they are blessed with the dubious privilege of a situation by Mortlake station and constant competition with the Waterloo to Richmond 'express'!

Phil Grant, the Micro Computer Centre's manager, with 20 years' computing experience to his name, has been running this company for 8 years, which makes it one of the oldest micro computer companies around. In fact, when the company was founded, Commodore itself consisted merely of a small office with a staff of four.

Small is beautiful

The Micro Computer Centre can hold the advice of a young engineer responsible for its birth. As a mini computer consultant on the verge of buying a new mini, Mr Grant was prompted, by the engineer helping him to transport his new purchase, to cross the barrier from minis to micros. Therein, he was told, lay the future. Mr. Grant, albeit somewhat ignorant of micros at the time, made the crucial move of trekking across to Commodore at Euston where he purchased a 2000 series Commodore micro computer. He never looked back. Within three months, his micro was achieving all he needed from a computer.

Having acquired four Commodore micros, he opened a small store with a notice firmly implanted in its window inviting passers-by to 'come in and play games' on his new micros. But this was merely a ploy to entice customers: Mr. Grant really had his eye to the business market and let it be known that business applications were also available. As is still the case in high street computer stores all over the country, Saturday mornings would see an infiltration of 'precocious computer buffs' implanting their sticky fingers on Mr Grant's keyboards — but help was at hand: escorting their offspring to this Commodore haven in darkest Sheen were their fathers — several managers of small businesses. One by one, bored with little Jimmy's prowess at zapping assorted creatures from outer space, they approached Mr. Grant seeking solutions to their various business problems. Mr. Grant's ability to prove to them that a computer could solve their problems resulted in an influx of orders and an escalation of business. And thus the Micro Computer Centre was born.

Soft-centred hardware store

Mr Grant's store is not merely a hardware retail outlet. Apart from the computers themselves, and various peripherals, the Micro Computer Centre offers a comprehensive range of business applications software — both in the shape of off-the-shelf packages and software penned by its own highly experienced programming staff. Mr Grant states that: "A large part of the company's success stems from writing our own software". When asked what sort of software they produce and for whom they cater, Mr. Grant succinctly replied: "Small applications for large businesses and large applications for small businesses". He insists that, in offering their own custom software, they are not out to exploit customers. If there is already something available on the market that will meet the customer's needs, Mr Grant will suggest he buys it. If not, the Micro Computer Centre's programming team will try to produce a piece of software suitable.

The Micro Computer Centre's success, so Mr Grant claims, lies in its emphasis on service rather than hard selling. He sees his company as a consultancy first and foremost; full support — training, maintenance, etc — is offered with all hardware and software sold. They have never advertised and

only recently employed a small sales force. Most sales come via personal recommendation.

The Micro Computer Centre's best selling product is their home-produced 'Office Mate', a fully integrated, modular office system containing, amongst other facilities, a purchase ledger, nominal ledger, sales ledger, cash book and stock control and, probably the greatest point in its favour, 283 modules which account for its flexibility. Mr Grant claims that their 'star performer' not only adapts to the customer rather than expecting the customer to adjust accordingly, but is also cheaper than its immediate competition.

I shall show by way of a couple of examples that the Micro Computer Centre, far from being mere high street traders, are also experts in the field of software development.



When some marketing personnel from a multinational drug company, having marketed a new product, asked Mr Grant for some help in analysing the visits of their representatives to doctors he was true to his word that his staff could do the job more efficiently than the 6-9 months it would take on a mainframe system. The marketing personnel visited Mr Grant on a Thursday: a skeleton of the program was ready the following Tuesday. The final result saw an analysis of visits to over 27,000 doctors stored into a Commodore 8032. "They said we couldn't do it", he boasted.

Featured recently on the BBC's series, "Their Life in Your Hands", was a program on research into cancer in women by the Institute of Obstetrics and Gynaecology at Hammersmith Hospital. With the assistance of Mr Grant and his team, statistics of hundreds of thousands of women have been stored on an 8032 Commodore with a 30M hard disc.

Why Commodore?

Until recently, the only hardware stocked by the Micro Computer Centre was Commodore born and bred. So why the initial choice of Commodore and why the recent digression?

When Mr Grant first took his giant step into the micro computer field, he believed that the only viable competition to Commodore was Apple. And Commodore, being cheaper, more flexible and more first-hand user oriented, got his vote of confidence. However, so far as sales of business micros are concerned, the immediate past has been a far cry from Commodore's initial success.

Mr Grant has strong views on Commodore's relative collapse in this field. He says of the 8000 series: "It was a very successful machine which led the market by far but sometime earlier Commodore had disbanded its R & D". He states that Commodore's designer, Chuck Peddle. . . went to Victor and designed Sirius; this should have been a Commodore machine".

He believes that, had Commodore's newest baby, the 8296D, been released in place of the somewhat disastrous 700 series, Commodore may well now hold the status in the business market which they deserve.

End of the tunnel

But, although sales of Commodore machines at the Micro Computer Centre have dropped, Mr Grant holds out a lot of hope for Commodore's return to the forefront of the business market. "I think they will come back early in '85 and when they do they'll come in with a bang". Commodore's reputation on home computers due to the resounding success of the CBM 64 has cost them some credibility in the business stakes. But Mr. Grant is convinced that: "The business market is going to be bigger than the home computer market". He believes that any machine which hopes to corner at least part of the business market needs to be IBM compatible due to IBM's dominance of the market with their PC.

As always, Commodore's price and promotion tactics will be right, he thinks, and as to the product itself: "I think the product will be good this time, although this could be pure optimism". But he has confidence in Commodore's vastly improved production facilities and new teams at work in Germany and the UK. With technical people of a different calibre now, there is every reason to believe that the product will be good.

"There is a place for the Commodore. If you look at the micro computers overall in the market. . .at the bottom end of the range there is no computer to compare with it."

So his predictions for '85 (in the various price categories) "Commodore at the bottom, Apricot in the middle and IBM at the top. I can't see anybody else getting a look in — I could be proved wrong".

Calling Commodore Retailers

If you own a small computer business or store where the emphasis is on Commodore machines or software, or if you know of anybody out there spreading the good name of Commodore through their trade, please let us know if you think they should be mentioned in *Your Commodore*.

BAND-AIDS



Chris Palmer gets stuck in to two music packages. Join him at the keyboards.

THERE CAN BE NO DOUBT THAT THE 64 boast one of the most powerful internal sound chips around. However, anyone who has delved into the workings of the SID chip will realise that its favour is not won easily. This leaves the way open for software writers to help the user win over SID. The two products we are looking at in this review exhibit two different ways of leading you towards a common goal. That is, making music on the 64, with the minimum of fuss.

One succeeds and the other fails. For the answer to this and other burning questions, read on!

Microsound 64

The Microsound package compromises a full size piano style keyboard, a tape or disc containing the driver software and a typewritten manual. The keyboard came as a very pleasant surprise for me. So often this sort of system will be let down by forcing you to play on a 'Mickey Mouse' keyboard. It has a span of a full four octaves and is cased in hard, black, ABS Plastic.

To the left of the keys are two sliders. They allow you to change certain variable components of the sound by moving them. Nice thought this, but why are they so stiff? On larger synthesisers this sort of arrangement is used to let the player put expression into a piece of music. To do this, the sliders have to be able to move under the slightest finger pressure to allow for both fast and subtle effects. Perhaps these sliders will loosen up a bit after use, but that leads me to wonder how good the contact will be.

The keyboard connects to the 64 through the two joystick ports. Once fitted the machine can be turned on and the software loaded. After selecting either disc or tape mode you are passed to the main screen.

It is from this screen that the majority of the sound creation and play characteristics are set up and altered. The bottom left part of the screen is devoted to the attack, decay, sustain and release levels for each of the three voices. The top of the screen shows which waveforms are active for the voices, along with filter routeing, sync and ring modulation. The

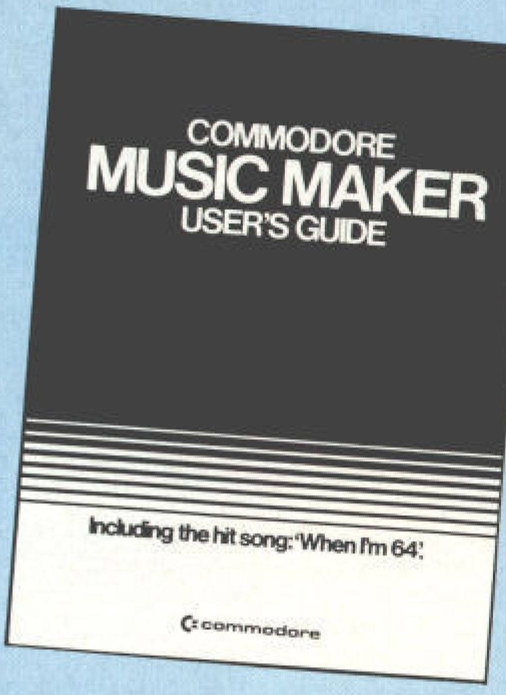
screen is completed by a filter status display and control key menu.

From this screen you can pass to menus which control tuning, patching and the sequencer. But, back to the main screen.

On power-up, the keyboard is set to monophonic mode with a fairly sharp brass sound. You can set the keyboard to polyphonic mode by toggling the '*' key. Because the 64 has only three oscillators, the polyphonic mode only lets you play three notes at a time. The software scanning of the keyboard seems to be a little slow, which makes fast playing a bit hit and miss. The quality of the sound depends very much on what output you're using. Through a TV speaker, even the best sounds seem rough, so putting the whole lot through a Hi-Fi improves things greatly.

Much as I admired the thoughts and sentiments behind the Micrsound 64, I could find very little to praise once I started to use it. For a start the manual is appalling! Even with my knowledge of synthesisers I found it quite tricky to understand the way some of the more advanced features worked. Confronting an inexperienced user with phrases like 'The instantaneous value of the output of oscillator 3 controls the pulse width' without any further explanation, is inexcusable.

Matters would have been helped if the disc contained some other sample sounds that could be loaded and examined. The manual didn't even show any example patches to play about with.



The software was slow to respond to the sound editing commands and I often found myself 'typing ahead', only to find that inadvertantly changed the wrong parameter and ruined the sound I was trying to create.

One of the most useful features of any music program of this sort has got to be the sequencer. Unfortunately, the Microsound sequencer was little better than the main part of the software. The most disturbing thing about it, was that you could get in, but you couldn't get out. That's right, the only way I found I could get back to the main menu was to break the program and RUN it again. Of course by doing this, you lose the sound you had created in the main menu, unless, of course, you had the foresight to save it. Even so, this is 1984 and software is supposed to be friendly, isn't it?

The sequencer will allow you to 'layer' three parts on top of one another and then play it back in unison. Sequences can be edited by running a cursor back to the relevant point and either deleting the note or playing the correct one over the top. Once you are happy with a sequence you can save it to your storage medium.

Despite my dislikes about this system, it does have some saving graces. If you want to find out how a synthesiser generates and controls sound, and are willing to persevere on your own, then the Microsound 64 could give you a lot of enjoyment. But if you expect to be able to sit down and play like Howard Jones or Rick Wakeman, then I don't think that the Microsound is for you.

There is one interesting development looming on the horizon that could turn the Microsound into a very attractive proposition for any amateur musician. On the back of the keyboard there is a 25 way D-type connector which is set aside for external triggering and digital sampling. Now, if Autographics can make a digital sampling add-on cheap enough, then I think the Microsound will become a popular addition to the set up of those of us who slave late into the night creating mini epics in our home studios.

Music Maker

It might seem strange to some people that having been slightly less than



complementary about the Microsound 64 with its proper keyboard, I am about to start raving about a system which uses what can only be described as a 'Mickey Mouse' keyboard. The article in question is, of course, the plastic, keyboard overlay which is supplied with Commodore's Music Maker package. To say that it wouldn't look out of place being given away free with a packet of cornflakes just about sums it up.

Its manner of operation is simple. The actual keyboard part of the unit is contained within a tough vinyl surround which fits over the body of the 64. When you press a key on the music keyboard, it in turn presses a key on the alphanumeric keyboard underneath. Now I must admit, when I ran across this at the PCW show, I discounted it as being too tacky to be of any real worth. But when you actually come to use it, you soon realise it works, and very well at that! You have not got the potential available that a full size keyboard offers you, but it's ideal for kids and anyone who hasn't progressed beyond the one finger stage.

The reason this stands head and shoulders above the Microsound is in the friendliness of the software. Each of the menus and sub menus are clearly laid out, so it's virtually impossible not to know what you are doing. The software also concentrates on getting people playing tunes as soon as possible, rather than dumping them in at the deep end, with only their own ideas to get them out.



The rest of the package also reflects this approach. As well as a very clear and concise manual, you also get an 'easy-play' song book. Contained in this are such classics as "Annie's Song", "The Birdie song" and the cringingly apt "When I'm 64". There is even a sheet of stickers with which you can plaster the keyboard if you have trouble finding A flat by the C (Ouch!)

Along, with the tunes for you to input yourself, the disc (or tape) also contains "When I'm 64", "Georgia" and the "Snow Waltz". All of which can be loaded and played immediately.

To the software

As I said earlier, it is the software which really makes this package stand out. Everything is menu driven and for the most part uses the function keys to select an option.

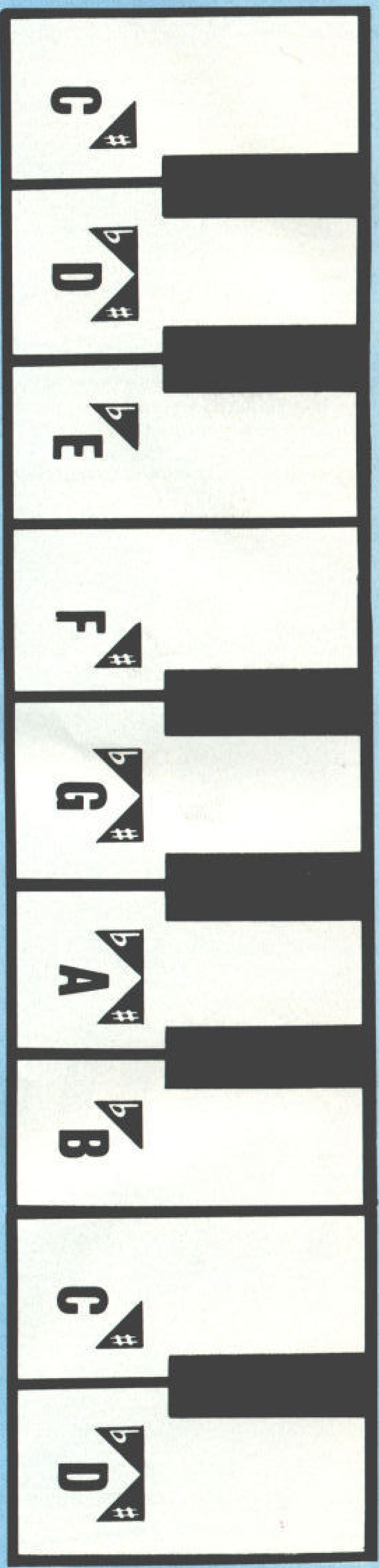
The software turns your 64 into what amounts to a Casio style keyboard. Along with the music voices from the keyboard, you are also offered an auto bass option and a choice of three drum patterns.

On power up, the software contains eight pre-defined voices, all of which are quite usable. If you want to change the voices you can divert to a sub menu where the sound parameters are displayed for you. An option to save and load the voice data is also provided, so it is possible to build up your own store of custom sounds. The manual also gives a table of values to provide a starting point for your own experiments.

The software also gives you the choice between monophonic and polyphonic play. Which, when you think about it, is no mean feat, considering that the Commodore keyboard isn't supposed to be able to be scanned that way. Needless to say, the authors are very pleased with themselves for doing this and unfortunately they wouldn't part with the secret.

The sequencer part of the program is split into two parts. Firstly, you input the notes by pressing the keys on the keyboard. The note you press is highlighted on a keyboard representation on the screen. You needn't worry about the timing of the piece at the moment, for this is the function of the second part. When you have completed the tune to your satisfaction, you then input the information for the timing of the piece by simply tapping on any key. The computer then stores this along with the tune. When you have finished, tapping the key again will play back the tune with the notes now in the proper time.

All told I found it a pleasure to work with the Music Maker package, despite the limitations imposed by the keyboard. The only depressing fact is that you can already buy song books to complement the system, featuring the works of Duran Duran, Michael Jackson and the like. No doubt at computer shows to come we will not be able to escape the strains of "Thriller" and "Rio". To combat this, I am already working on some transcriptions of Emerson, Lake and Palmer's most inaccessible moments for just that occasion. Let battle commence!





Les Allan explains how to protect your valuable programs from prying eyes.

OK! SO YOU'VE PRODUCED a masterpiece of software that you are fairly pleased with, maybe using some techniques that you would like to keep up your sleeve. Well, how do you go about protecting software? The truth of the matter is that nothing is capable of being fully protected. All that can be done is to deter or slow down the way in which the 'pirate' can get into your program. How do you start if you know little or nothing about the subject or indeed the areas of the computer's memory that are equipped to deal with this problem. The following routines can be used either separately or together in order to provide the degree of protection required.

(1) Use of screen

In order to read from the screen to check the name file of the program being loaded it is necessary to provide a method of ensuring that the name file is always located in the same part of the screen. This is achieved by using the CLR (reverse field heart) at the beginning of the name file thus:

```
SAVE"(CLR)NAME FILE"
```

The protection routine can then be used to read from the screen thus:

```
10 IFPEEK(1024)<>14THEN SYS64738

1024......... screen location
14........... poke code for N
64738.. cold start BOOT code
```

(2) Use of the cassette buffer

The maximum number of characters that can be displayed to the screen for the name file from the tape being loaded is 16. However, it is possible to have a name file of up to 76 characters in length where the hidden characters can be used in the protection method thus:

```
SAVE"(CLR)NAME FILE
PROTECTION ROUTINE"
```

When loaded from tape all that is displayed is the name file but the complete contents of the save are located in the cassette buffer and can be used thus:

```
20 IFPEEK(850)<>80THENSYS 64738

850.................. location in cassette buffer for P
80................................ ASCII code for P
64738....................... COLD start BOOT code
```

(3) Disable keyboard and/or list save

```
30 POKE775,200:REM DISABLE LIST COMMAND:POKE 775,167 TO RE-ENABLE
40 POKE808,225:REM DISABLE RUN/STOP :POKE 808,235 TO RE-ENABLE
50 POKE818,32 :REM DISABLE SAVE ROUTINE:POKE818, 237 TO RE-ENABLE
```

These functions, once the program has been loaded and run, will prevent the program from being stopped and broken into.

(4) Move basic

Bottom of BASIC is normally located at dec. 2049 but can be relocated by use of the following pointers:

```
10POKE43,lo byte and 255
20POKE44,INT(hi byte/256)
```

By adding REM BACK DELETE statements the change can be disguised and the 'pirate' misled into believing that the program is machine code based thus:

```
10 POKE43,0:POKE44,20: RUN:REM"(INST★27)
SYS4096
```

In order to achieve this add REM" " and move the cursor back to the right hand quote marks. Then make a number of INST followed by the same number of DEL. When the program is subsequently listed the line looks thus:

```
10 SYS4096
```

In order to make use of this facility the program must first be written and saved in the normal area of BASIC with the screen, cassette buffer and disable functions incorporated. Now carry out the following steps:

(1) SYS64738 to clear memory

(2) Add new line 10 to change bottom of memory and run program

(3) POKE 43,0:REM SET UP LO BYTE FOR BOTTOM OF MEMORY

(4) POKE44,20:REM SET UP HI BYTE FOR BOTTOM OF MEMORY

(5) LOAD in program

(6) POKE43,1 : REM RESTORE LO BYTE FOR BOTTOM OF BASIC

(7) POKE43,8 :REM RESTORE HI BYTE FOR BOTTOM OF BASIC

(7) SAVE"(CLR)NAME FILE PROTECTION ROUTINE"

(5) LOAD M/C FILE FROM BASIC

Machine code files can be loaded from BASIC to reduce the overall time taken to load the program from tape and also to relocate the program such that it is transparent to the user. Because of the way BASIC is configured it is necessary to load in this manner :

```
10 IFA=0THENA=1:LOAD" M/C FILE 1",1,1
20 IFA=1THENA=2:LOAD" M/C FILE 2",1,1
```

(6) M/C BOOT LOAD

This topic is outside the scope of this article but for those of you adventurous enough this provides the best method of protection as the programs saved in this way are usually below the bottom of BASIC and are difficult to stop particularly when the warm start vectors are set up at dec. 770 and 771 to point automatically to the start address of the machine file.

Garry Marshall puts your programming prowess to the test in this new series for the CBM 64.

PROGRAMMING PROJECTS

THIS NEW SERIES WILL present a programming project in the form of a problem to be tackled with the aid of the computer. The projects have been devised not only to call on programming techniques that are interesting in a general way, but also to show off the best features of the 64. Graphics, sound production and sprites will be prominently featured. Some of the programs that are necessary to carry out the project will be developed. The intention is to introduce some methods for solving problems, to introduce some useful programming techniques, and to lead you into further explorations associated with the project and with the 64.

We begin this month with a 'turtle chase'.

The Project

There are a number, N, of turtles on a floor. They are numbered from 1 to N. Their positions can be stored by placing the X- and Y-coordinates of each in the relevant elements of arrays named X and Y. The turtles move so that each pursues the next one: turtle 1 pursues turtle 2, turtle 2 pursues turtle 3, and so on, with turtle N pursuing turtle 1. They all head directly towards the turtle that they are pursuing, and they all move *at the same time*. This situation is most conveniently simulated by letting all the turtles move at discrete instants, as on the ticks of a clock. At each instant, every turtle moves a fixed distance towards the position occupied by the turtle that it is pursuing. Figure 1 shows the first few movements for a group of five turtles.

The program should begin by accepting a value for the number of turtles, N. It can also accept their initial positions, but symmetric patterns will result if the turtles are systematically placed, perhaps in positions that are equally spaced around a circle.

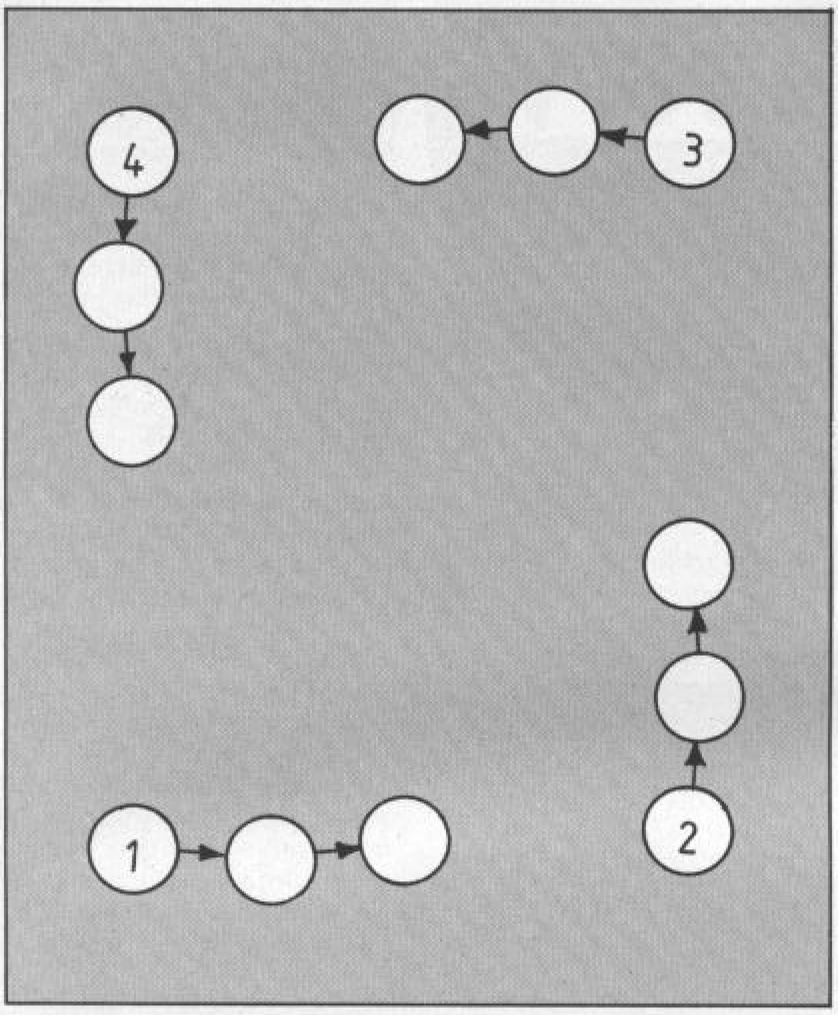


Figure 1. Initial steps for 4 turtles

The solution

Because the turtles all move at the same time, we shall not only need the arrays X and Y to hold the current positions of the turtles, but also two further arrays to hold the positions that they move to at each instant in the simulation. If we do not use a second pair of arrays, but directly update the arrays holding the current positions with new positions then some of the turtles will not head for the correct positions when they move. This would mean that we should not be simulating faithfully the situation in which all the turtles move simultaneously. It is also convenient to give all the arrays a dimension of N+1, for then the contents of the first elements of the arrays X and Y can be copied into the (N+1)th and the case of turtle N chasing turtle 1 can be dealt with in the same way as all the others.

Figure 1 should be sufficient to show that all the turtles will spiral in towards each other as they move. (If this is not easy to see, then carry out a few more steps with a pencil and paper simulation.) Starting the turtles off from positions that are equally spaced round a circle gives a neat symmetrical pattern for their paths. But if you are not particulary worried about neat effects, any starting positions will do, and the result will still be a pattern of spirals. In our program, the turtles are spaced around a circle with its centre at the centre of the screen and having a radius of 90 dots. This gives us the first part of the program for accepting the number of turtles and for setting up their initial positions (see below)

The gap between lines 60 and 110 will be filled in a moment.

We can now write the program segment for computing the successive positions of the turtles. The steps to be taken by a turtle towards the turtle it is pursuing are five screen dot positions in length. The new position for the pursuing turtle is calculated by finding the difference between the x-coordinates and the difference between the y-coordinates for each turtle and its prey. The distance, D, between them can then be calculated by using Pythagoras' theorem.

The new program can then be calculated by reducing the differences in the x-coordinates and in the y-coordinates in the same proportions so that the total change in distance is five. When all the new positions have been computed the X and

```
10 INPUT "NO. OF ROBOTS";N
20 DIM X(N+1),Y(N+1),XN(N+1),YN(N+1)
30 FOR K=1 TO N
40 X(K)=128+90*COS(6.28139*K/N)
50 Y(K)=96+90*SIN(6.28139*K/N)
60 NEXT K
110 X(N+1)=X(1): Y(N+1)=Y(1)
```

Y arrays can be updated with these values and then the process can be repeated. The program is stopped when the turtles are within five steps of each other. (They cannot very well take a step towards each other if they are already closer than that!) The instruction for stopping is written as
180 IF D<5 THEN 180
rather than as
180 IF D<5 THEN STOP
so that the display is not spoilt when the program halts. This program segment is:

```
150 FOR K=1 TO N
160 A=X(K+1)-X(K): B=Y(K+1)-Y(K)
170 D=SQR(A*A+B*B)
180 IF D<5 THEN 180
190 XN(K)=X(K)+INT(5*A/D)
200 YN(K)=Y(K)+INT(5*B/D)
210 NEXT K
220 FOR K=1 TO N
230 X(K)=XN(K): Y(K)=YN(K)
240 NEXT K
```

Now, although these program segments carry out the simulation, they are not very exciting in themselves. By adding some graphics we can display the successive positions of the turtles and see the paths that they take building up as they approach each other.

The first steps that are necessary before any graphics can be displayed are to set the 64 to its bit-mapped high-resolution graphics mode, to position the bit-mapped display in the memory, and then to clear it and set its foreground and background colours. This is done by placing the following lines in the gap we left earlier:

```
70 POKE 53272,PEEK(53272)OR 8
80 POKE 53265,PEEK(53265) OR 32
90 FOR I=8192 TO 16192:POKE I,0:NEXT I
100 FOR I=1024 TO 2023:POKE I,22:NEXT I
```

The position of a turtle can be shown by plotting a dot at its current position. A single dot can be plotted in row R and column C by:

```
1000 RO=INT(R/8): CO=INT(C/8)
1010 L=R AND 7
1020 BIT=7-(C AND 7)
1030 BYTE=8192+RO*320+CO*8+L
1040 POKE BYTE, PEEK(BYTE) OR 2↑BIT
1050 RETURN
```

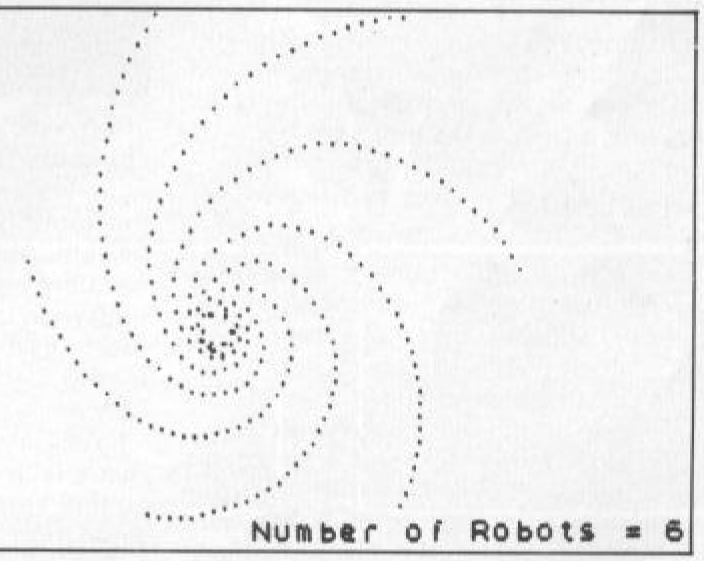
Putting all this together gives us the following complete program. We have added lines 120 to 140 to convert, in turn, each of the turtle positions to the form required by the plotting subroutine, and then to call the subroutine itself. Line 250 is also added so that as soon as one set of positions is plotted, the program goes back to line 110 to compute the next positions and to plot them.

```
10 INPUT "NO. OF ROBOTS";N
20 DIM X(N+1),Y(N+1),XN(N+1),YN(N+1)
30 FOR K=1 TO N
40 X(K)=128+90*COS(6.28139*K/N)
50 Y(K)=96+90*SIN(6.28139*K/N)
60 NEXT K
70 POKE 53272,PEEK(53272)OR 8
80 POKE 53265,PEEK(53265) OR 32
90 FOR I=8192 TO 16192:POKE I,0:NEXT I
100 FOR I=1024 TO 2023:POKE I,22:NEXT I
110 X(N+1)=X(1): Y(N+1)=Y(1)
120 FOR K=1 TO N
130 C=X(K): R=Y(K): GOSUB 1000
140 NEXT K
150 FOR K=1 TO N
160 A=X(K+1)-X(K): B=Y(K+1)-Y(K)
170 D=SQR(A*A+B*B)
180 IF D<5 THEN 180
190 XN(K)=X(K)+INT(5*A/D)
200 YN(K)=Y(K)+INT(5*B/D)
210 NEXT K
220 FOR K=1 TO N
230 X(K)=XN(K): Y(K)=YN(K)
240 NEXT K
250 GOTO 110
1000 RO=INT(R/8): CO=INT(C/8)
1010 L=R AND 7
1020 BIT=7-(C AND 7)
1030 BYTE=8192+RO*320+CO*8+L
1040 POKE BYTE, PEEK(BYTE) OR 2↑BIT
1050 RETURN
```

When this program is run, it first asks for the number of turtles. When this number has been entered, the program then takes a minute or two to set the high-resolution mode and to clear the relevant section of memory. You can 'see' this happening on the screen, but do be patient.

The display produced by this program for N=6 is shown in Figure 2. If the stopping condition is removed by deleting line 180, some quite spectacular effects are produced. The effect results from the turtles oscillating about each other when they are closer than five dots as they continually overshoot the turtle they are chasing.

Figure 2



Further developments

The program that is developed here contains the essence of the solution to our problem, but there are many ways in which it can be varied. Among these ways are:

Changing the starting positions of the turtles.

Changing the length of the steps taken by the turtles.

Changing the order in which the turtles pursue each other.

Changing the rules for what happens when two turtles meet.

Perhaps rather than 'passing through' each other they should bounce off each other.

Exploring the 'flare' patterns that result from different numbers of turtles, different step lengths and different 'collision' rules.

Using sprites for the turtles and moving them rather than just marking the paths of the turtles.

No doubt you will discover many other ways of adapting and developing this project.

Steve Ireland talks to his computer in morse. And it talks back.

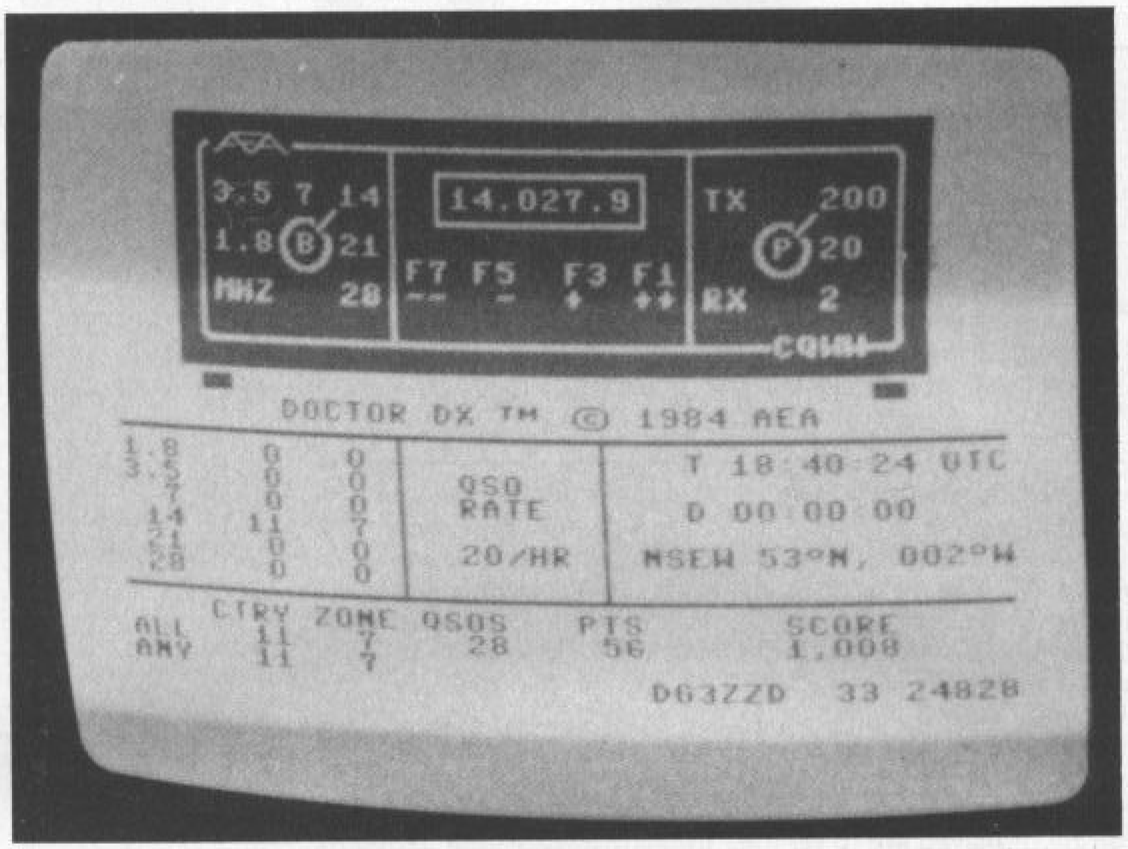
ON THE AIR

CALLING ALL COMMODORE 64 owners. If you are an amateur radio enthusiast then an American company called AEA has just invented *the* computer game for you. It is called Doctor DX and comes in the form of a cartridge which plugs into the expansion port of the 64.

Around the world

Let me explain a little of the background to the game for those of you not familiar with the world of amateur radio. During the course of a year there are a number of contests for radio amateurs who are keen CW — morse to the uninitiated — operators on the shortwaves. These are worldwide events and stations from Bognor to Bangkok, Trowbridge to Tahiti and Sheffield to Siberia will participate. The idea of the contest is to contact as many stations as possible in as many countries as you can find over a twenty-four hour period. Radio amateurs erect huge aerials and go to 'rare' (ie, rarely operated from) countries in order to win these contests and the fame and acclaim which accompanies them. Many of them also spend large sums of money in these endeavours.

If you don't have a huge aerial and can't afford the ticket to a Pacific Isle, you can conjure up a prize winning station and a contest to go with it simply by plugging a Doctor DX into your 64.



On screen

Once it's up and running, a graphic representation of a radio transmitter and receiver appears on the screen. The 'transceiver' covers the six most popular amateur wavebands. There is a phono connector at the back into which you must plug either a hand key or an electronic keyer in order to transmit your morse 'messages'. Having done this, you are invited to enter your chosen latitude and longitude co-ordinates (you may operate from any country in the world) and the time in GMT at which you wish to begin. Then, on pressing RETURN on the 64, Doctor DX proceeds to simulate suitable radio conditions for your location over a 24 hour period.

Wave forms

Radio conditions vary a lot over a 24 hour period depending on where you are in the world and the time of day. This is because of the varying conditions of the ionisphere. Long distance shortwave communications rely on the reflection of radio waves by the ionisphere. To do this, the ionisphere needs to be ionised to a certain degree, depending on the frequency of the radio waves, and this ionisation is caused to some considerable extent by the position of the sun relative to the frequency. A little complicated, isn't it!

Well, within the 16K of EPROM memory in the Doctor DX all of these factors are simulated. I found the simulation quite remarkable and I've been 'on-the-air' for some 14 years. Stations appear on the bands and can be communicated with and you can also overhear stations communicating with each other. All of this is just like real (contesting) life. When you are operating in the lower band segments, the stations are transmitting at much faster speeds and the operators are more polished. This, too, is also often the case in real life. •

Mastering morse

Of course, the Doctor DX is designed for the specialist user; not everyone knows morse or wants to learn it. However, for the enthusiast it provides an excellent morse trainer and anyone who wants to learn can use the Doctor DX once they have reached about 5 words per minute. It also allows the rest of us to take part in competitions from anywhere in the world without spending a penny on airline tickets or to 'transmit' with no other equipment than a Commodore 64.

So, if you've been considering becoming a ham radio enthusiast but have been put off by the cost of the equipment then the Doctor DX will give you an idea of the excitement to be had from talking to the rest of the world. (Of course, you have to have *some* morse expertise first!) Sitting in front of the screen listening to the noise on the airwaves, I could believe that I was talking to someone on the other side of the world and not just to the computer. Surely there can be no better recommendation for a computer simulation than this.

Doctor DX costs £96.95 and is available from ICS Electronics, PO Box 2, Arundel, West Sussex.

Steve Ireland is the editor of Ham Radio Today.

We've got the TALENT



and our games prove it!

Amazing graphics, fast and furious action, challenging strategy, compelling adventure – this first wave of games from TALENT has got the lot!

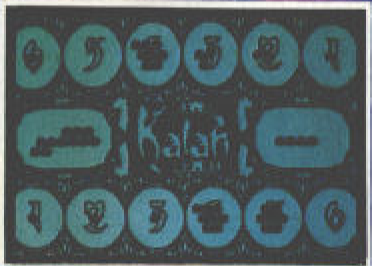
Written by professional computer scientists using powerful new programming techniques (which leave machine code standing), these games have pushed home micros to the very limit.

Kalah

ARCHIPELAGO













KALAH

One of the oldest games in the world! Originally played by Bedouins using pebbles and holes scooped in the desert sand, this strategy game has absorbed the mind of man for thousands of years. Superb graphics and music, with a choice of levels and depth of search. Easy to learn, fiendishly difficult to beat!

Commodore 64 Cassette £7.95 Disk £9.95

LASER REFLEX

A new angle in alien zapping! Against a brilliant backdrop of stars, you must defend your asteroid base from wave after wave of swooping mutating starships. Your defence – a powerful laser deflected against a giant mirror. How long can you hold out before your defences crumble?

BBC Model B Cassette £7.95
Electron Cassette £7.95

ARCHIPELAGO

Dare you venture into the ancient undersea labyrinths which link the islands of the Archipelago? Fabulous jewels await you there – but the guardians of the treasure mark your every move, waiting to deal a swift and terrible vengeance. How long can you survive?

A fast-moving, action-packed game with brilliant graphics, animation and music.

Commodore 64 Cassette £7.95 Disk £9.95
(joystick needed)

WEST

A compulsive adventure game set in a ghost town in the Wild West. Your task – to flush out a gang of bank robbers and retrieve their loot – with only your trusty steed and your six-shooter to aid you!

Test the speed of your draw, your skill at cards, your powers of survival!

The first of a stunning series of adventure games from Allan Black, played in real-time.

Commodore 64 Cassette £9.95 Disk £12.95
BBC Model (text only) Cassette £7.95
Electron (text only) Cassette £7.95



PANORAMA (H)

An outstanding graphics package for the Commodore 64. Features include: free-hand sketches, circles, ellipses, block move, copy and reverse, magnification choice of cursor, joystick or key control. PLUS texture definition and texture and colour fill. PANORAMA (H) uses the full colour facilities of the Commodore 64.

Commodore 64 Cassette £17.95 Disk £19.95
The programme is menu-driven and comes with a comprehensive illustrative instruction booklet.

TALENT

Curran Building, 101 St James Road
Glasgow G4 0NS Tel 041 552 2128

SOFTWARE FROM SCOTLAND

Talent games should shortly be available from your local software retailer or post off the coupon now.

INSTANT CREDITCARD 24 HOUR ORDERING SERVICE TEL 041 552 2128

TO: TALENT COMPUTER SYSTEMS FREEPOST (no stamp required) GLASGOW G4 0BR
Please send me the following items

COMMODORE 64	Quantity		
PANORAMA (H)	—	Cassette	(£17.95) ☐
PANORAMA (H)	—	Disk	(£19.95) ☐
ARCHIPELAGO	—	Cassette	(£ 7.95) ☐
ARCHIPELAGO	—	Disk	(£ 9.95) ☐
KALAH	—	Cassette	(£ 7.95) ☐
KALAH	—	Disk	(£ 9.95) ☐
WEST	—	Cassette	(£ 9.95) ☐
WEST	—	Disk	(£12.95) ☐

BBC MODEL B	Quantity		
LASER REFLEX	—	Cassette	(£ 7.95) ☐
WEST (text only)	—	Cassette	(£ 7.95) ☐
ELECTRON			
LASER REFLEX	—	Cassette	(£ 7.95) ☐
WEST (text only)	—	Cassette	(£ 7.95) ☐

Total (includes postage and packing)

I enclose a PO/Cheque to the value of £ ____________

Please debit my ACCESS ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Name (block capitals) ____________

Address ____________

____________ Post Code ____________

Signature ____________ Tel ____________

Please allow up to 28 days for delivery. We shall not bank your remittance until your order has been despatched.
Dealer enquiries welcome. Contact Veronica Colin on 041-552 2128.

Commodore's Plus/4 contains a suite of four programs. Does this make it a credible business machine? Bob Wallace answers this question.

PLUS/4 SOFTWARE QUARTET

IN A YEAR OF UPS AND DOWNS IN THE home micro market, there have been some huge success stories and tales sorrowful enough to bring tears to a CPU's eyes. In a field with such a competitive spirit, all manufacturers are trying as hard as possible to make their machines as versatile and powerful as they can, whilst also keeping the price of the machine as low as possible.

Nearly all extra packages for a home micro used to be very expensive pieces of software. Now some of these packages come as standard, with at least one being incorporated in the more expensive MSX machines.

Not to be outdone by the rapid Japanese invasion, Commodore have rallied forth and produced probably their most adventurous machine: the Plus/4.

It does not contain one software package, but a suite of four programs all resident in ROM. The packages are; a Word Processing package; a Spreadsheet package; a Database system; a Business graphics package.



Word Processing

The first program to be dealt with is the Word Processing package. For those who are unfamiliar with such things, it is a program that allows the user to type out letters and documents directly to the TV or monitor screen, modify the text to a desired standard and then produce a hard copy printout via a printer.

Once a document has been printed the information can be stored on tape or disc for future use.

To enter the word processor, you just press the F1 key and that's it. The computer keyboard is now acting as a typewriter keyboard, so what you type is what appears on the screen. The idea behind such a package is that errors are easy to correct. This is certainly the case on this processor. Using the computer's text editing facilities, one can easily insert missing letters and delete letters that you have typed in wrongly.

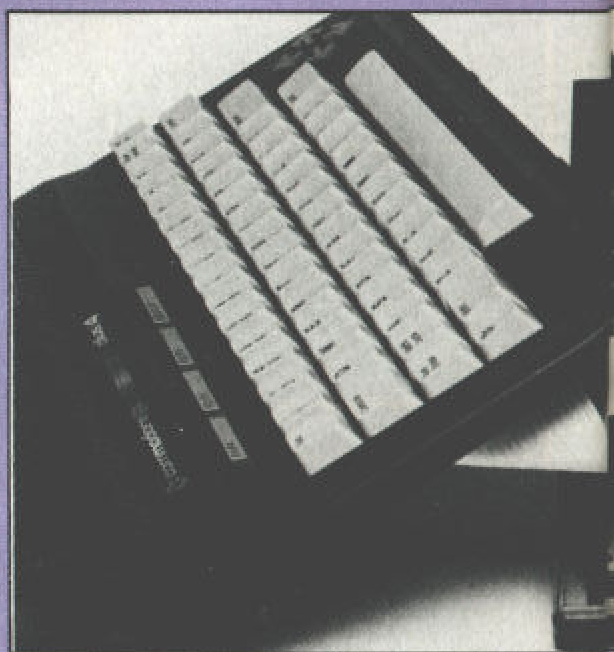
The processor allows a maximum line length of 77 letters with auto wraparound to prevent split words appearing at the end of lines. The 77 letters are fitted on to a 40 column screen by scrolling the screen to the left when the forty columns are filled. This I found to be rather distracting when typing in large documents. One remedy would be an 80 column display, which is not possible at present on the Plus/4.

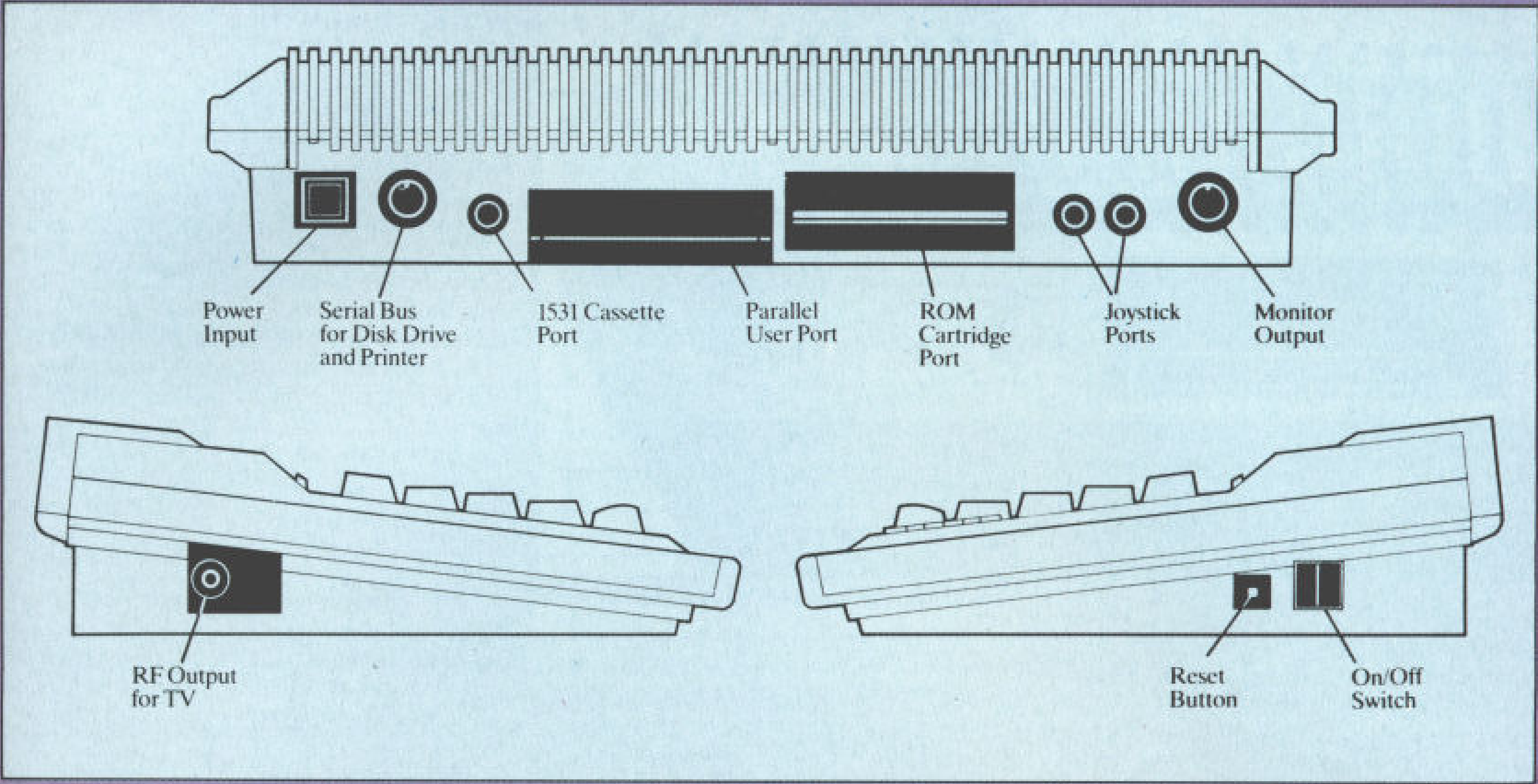
Having entered a document there are a host of inbuilt commands. These enable the user to insert or delete lines, edit and move blocks of text around within the document, set page length and size. The very nice touch is the search and replace command which allows the user to search for a given string and change that string to another specified by the user.

Other commands available to the user include the ability to set Tabs, Centre text lines, Right justify text, Merge files, Load and Save Files to disc, all of which help to produce a professional looking document.

Having said that, the quality of the document is not finally decided by the processor, but by the printer. If a cheap Dot Matrix is used then a poor quality document will be the obvious result.

Commodore have tried to give the user as many commands as possible to ensure the easy use of the processor. All commands entered are by a simple mnemonic format. With a few days use they should be easy to remember and they allow full use of the facilities available. In conclusion, I feel that the package has great potential for the small business user or club which needs fairly simple documents producing on a regular basis. If however you need to produce very lengthy documents, this processor is not ideally suited.





The Spreadsheet

At first, I thought that reviewing a spreadsheet would be about as interesting as watching paint dry. I soon realised that the documentation supplied by Commodore made entering data to the spreadsheet very simple and almost a pleasure.

What exactly is a spreadsheet? For me this question needed to be answered in one way only: What could it do for me?

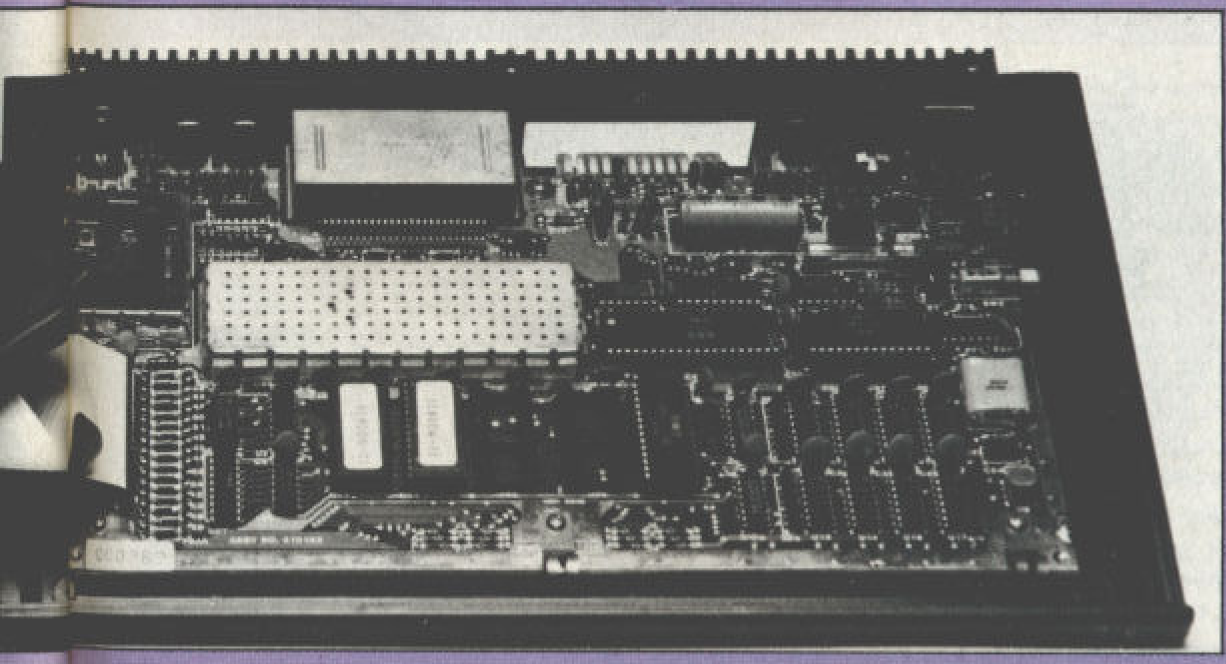
It soon became apparent that if I was willing to spend a little time entering data about my income and expenditure, I could use the computer to budget my account and even get a neat printout showing where I'd spent it all. Fortunately my printer does not print in red so it did not look so bad.

To enter the spreadsheet, you must first enter the word processor and then in command mode type TC which means 'to calculations', and then you enter the spreadsheet. The spreadsheet is organised as 17 columns and 50 rows, of which only 3 columns and 12 rows are visible on the screen at any one time. To compensate for this a scrolling screen is used, (as in the word processor).

The intersection of a row and column is called a cell and a large cursor block indicates the current cell location. By pressing F2 the cell is moved to the right, F1 moves the cell to the left. In the cells one can enter either numeric or string data, so you can have meaningful column headings on the final printout. Financial date can be edited and modified with a reasonable degree of ease. The entering of data is fairly straightforward and with a little practice, you should soon become quite proficient at producing a good printed copy.

The package also allows the use of formulae from within the spreadsheet, so that quite complex calculations such as interest rates and mortgage repayments can be calculated. The method of entering formulae is a little complex and needs some time before it is mastered. As with the word processor, any file that is created can be loaded or saved from disc. The obvious way to evaluate the uses of this package is to try it yourself and hopefully it will meet the standards you desire.

I feel that this package is only going to be of use to the small businessman or club which deals with limited financial transactions. To the average computer buff, it may seem a bit of a waste of ROM space.

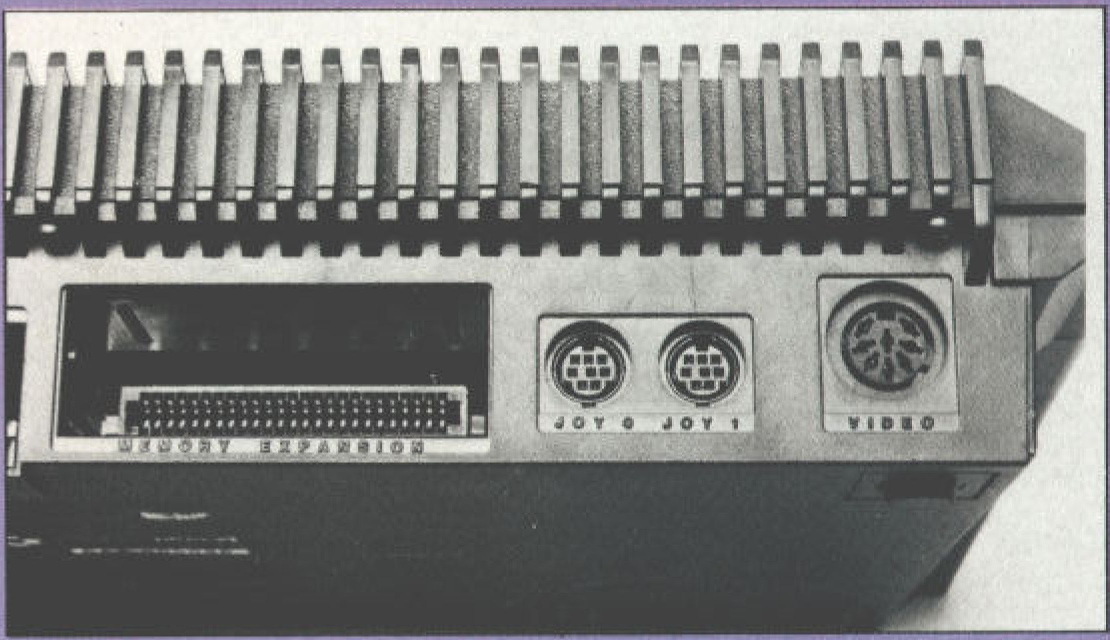


The Database

The uses of a database are only limited by the imagination. A good database will allow the user easily to enter a given set of facts, and, at a later date retrieve them for further use. Applications could include listing names and addresses, personnel files, collections, recipes etc.

Under normal circumstances, one would possibly keep this type of information in some type of card indexed system. The Plus/4 database puts an end to having to plough through endless card systems to retrieve your precious data.

To enter the database you first enter the word processor and type TF + RETURN. Before actually entering any data it is advisable to plan your file on paper first as this will speed up entry to the



computer.

A database stores information in Fields which you specify. Thus for a telephone directory, one might need 5 fields to cope with first and last names, address (two fields) and telephone number. The package allows up to 17 fields, each of which can contain 38 alphanumeric characters per field.

The maximum number of records per file is 999. This may be a little restrictive if the package were to be used, for example, by a school as a pupil database.

Once you have defined your fields the entry of data is very simple and a novice should have little or no problems with this aspect of the package.

As soon as the required data has been entered into the computer, several useful commands are available for the user to manipulate the data.

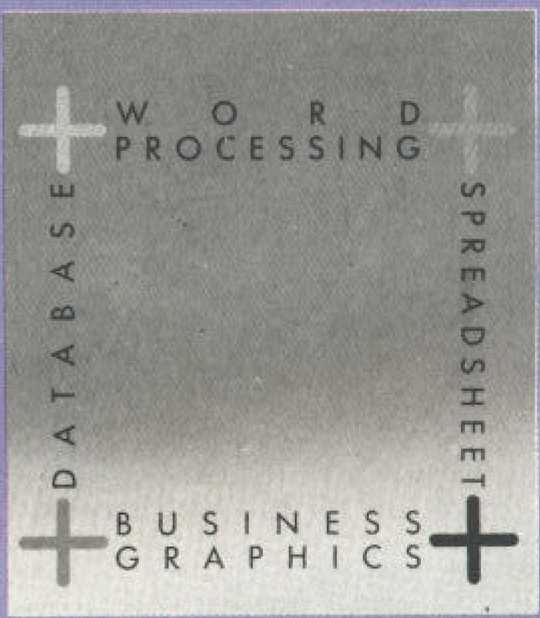
One of the best features of a database is its ability to sort data into a specified order, such as alphabetically, or numerically. This package allows this, but it uses memory occupied by any data left in the word processor so caution should be exercised when doing a sort.

The package allows the user to produce a printed report on the data which is stored in the database, but to do this requires the use of several entries from the keyboard. The number of user inputs could have been cut down a little and the hard work left to the computer.

If you do not require a printout of the data then the two commands used most frequently are Review and Search.

The review command flips very quickly through the data stored starting at any specified record number. When the desired piece of information is in view, pressing CONTROL and S will freeze the display, pressing RETURN re-enables the review.

If a full review is not wanted, one can enter the search mode where the computer will ask for a string or number to search for and then display all records that have that particular string in them.



This is a most useful command.

The data base is by far the best package of the four offered by the Plus/4. If you have any kind of hobby or job that needs quick and easy access to information, then you need a database.

Business graphics package

This package is not really a separate program, but an extension of the spreadsheet. It enables the user to output numeric information in graphic format.

I was somewhat disappointed to find that even though the Plus/4 has extensive Hi Res graphic capabilities, they are not used in this program. The excuse is that the characters must be printable by all Commodore printers. Since even very cheap printers have Dot Programmable characters, I feel that the excuse is a little weak.

The entry of data is very straight forward, and a nice touch is that the package has an auto scaling function which keeps the graphs to a readable scale.

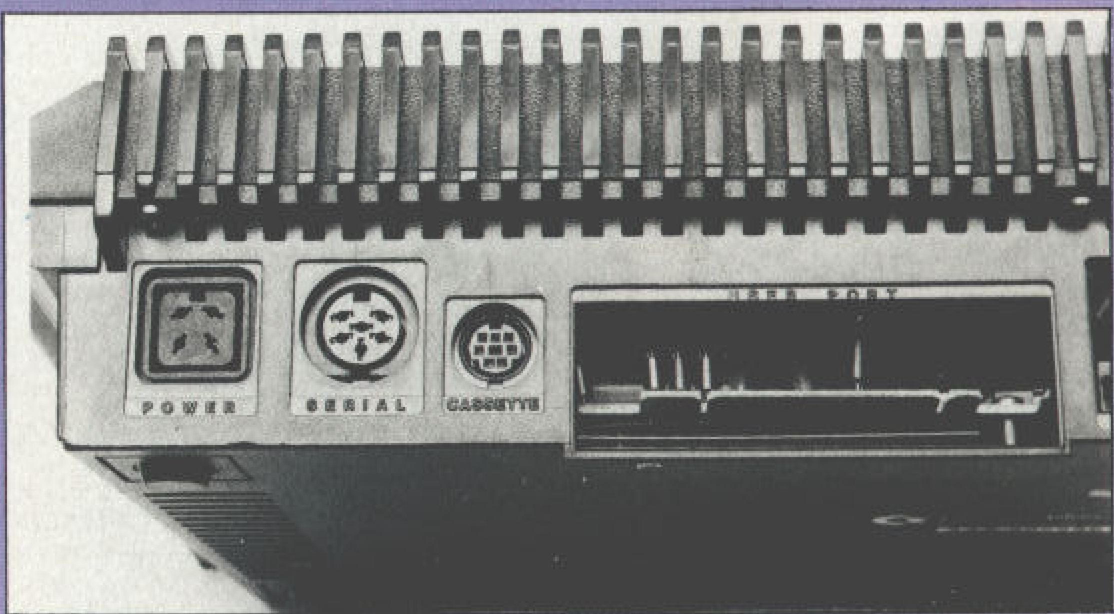
The graphics can be dumped to a printer and added to any document or spreadsheet. The quality of the graphs is not as good as it would be in Hi Res, but one poor graph is better than no graph at all.

Bar graphs and point graphs are catered for, though the bar graphs are more pleasant to look at.

All four

Commodore have produced a relatively cheap micro with a lot of the facilities normally found on much more expensive machines. However, the packages offered are probably only useful to a limited number of people. If you are considering buying a Plus/4 for the built-in software make sure that it will cope with all the tasks you want it to perform, not just now but also in the foreseeable future. Otherwise you may be frustrated by its limitations.

However, within those limitations the Plus/4 and its software is a fairly successful package to which I would offer a qualified welcome.



If you have something to sell now's your chance! Don't turn the page — turn to us!
Rates of charge: 40p per word per issue (minimum of 15 words).Please state classification and post to YOUR COMMODORE CLASSIFIED DEPT., 1 GOLDEN SQUARE, LONDON W.1.

Please place my advert in YOUR COMMODORE . . for . . . . . . issues commencing as soon as possible.

I am enclosing my Cheque/Postal Order/International Money Order for: (delete as necessary) £. . . . . (Made payable to A.S.P. Ltd)

OR Debit my Access/Barclaycard (Delete as necessary)

All classified advertisements must be paid for in advance.

Please use BLOCK CAPITALS and include post codes.

Name (Mr/Mrs/Miss/Ms) ....
(delete accordingly)

Address..............................

....

....

Signature.............................. **Date**..................

Daytime Tel. No. ....

AD INDEX

MAIL ORDER PROTECTION SCHEME

If you order goods from Mail Order Advertisers in this magazine and pay by post in advance of delivery, this publication will consider you for compensation if the advertiser should become insolvent or bankrupt, provided:

1. You have not received the goods or had your money returned; and
2. You write to the publisher of this publication explaining the position not earlier than 28 days from the day you sent your order and not later than 2 months from that day.

Please do not wait until the last moment to inform us. When you write, we will tell you how to make your claim and what evidence of payment is required.

We guarantee to meet claims from readers made in accordance with the above procedure as soon as possible after the advertiser has been declared bankrupt or insolvent to a limit of £1,800 per annum for one advertiser, so affected, and up to £5,400 p.a. in respect of all insolvent advertisers. Claims may be paid for higher amounts, or when the above procedures have not been complied with, at the discretion of this publication, but we do not guarantee to do so in view of the need to set some limit to this commitment and to learn quickly of reader's difficulties.

This guarantee covers only advance payment sent in direct response to an advertisement in this magazine (not for example, payments made in response to catalogues, etc, received as a result of answering such advertisements)

CLASSIFIED ADVERTISEMENTS ARE EXCLUDED.

YOUR COMMODORE

Lineage: 40p per word.

Semi display: £9.00 per single column centimetre
Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01-437 0699

Send your requirements to:
SALLY COLLINS
ASP LTD, 1 GOLDEN SQUARE,
LONDON W1

CATALOGUES

NUMBER ONE FOR CHOICE

The Six-Four Supplies Company is the premier independent mail order software specialist for the Commodore 64.

Our new catalogue of over 1500 items of recreational, business and educational software is now available. This is free to customers, 50p plus large SAE to others.

The catalogue is packed with product details in all categories and is the most thorough compilation of its type.

Add your name to our mailing list NOW for details of all our special offers — and look for our full-page advert in the Commodore magazines.

Most orders are dealt with immediately — use our hotline to test this out! And let us help and advise you if you're stuck for choice or have a problem which needs sorting out. Try us once — and you'll stick with us!

The Six-Four Supplies Co.,
PO Box 19, Whitstable,
Kent CT5 1TS.
Access/Visa: (0227) 266289, 462600

SOFTWARE

ZOOMSOFT

COMMODORE 64 SOFTWARE SPECIALISTS

Title	cass	disc
Boulder Dash	8.95	10.95
Bristles	8.95	10.95
Astro Chase	8.95	10.95
Flip Flop	8.95	10.95
The Hulk	9.95	13.95
Fort Apocalypse	9.95	14.95
Aztec Challenge	8.95	12.95
Kong Strikes Back	7.90	—
Slinky	8.95	12.95
Zork I	—	11.95
Zork II	—	11.95
Zork III	—	11.95
Starcross	—	11.95
Deadline	—	11.95
Encounter	9.95	12.95
Flight Simulator	34.25	37.95
S.A.M.	—	41.95
Forbidden Forest	8.95	12.95
Jetboot Jack	7.94	—
Nato Commander	9.95	12.95
Sorceror of Claymorgue Castle	9.95	13.95
Trollie Wallie	6.95	8.95
Daley Thompson's Decathlon	7.95	—
Football Manager	7.95	—
Super Base 64	—	86.25
Juice	—	29.95
Adventure Creator	ROM	29.95
Oil's Well	ROM	26.95
Beyond Castle Wolfenstein	—	21.95

Title	cass	disk
Castle Wolfenstein	—	27.95
Master Type	—	32.75
Sprite Maker	9.95	—
Stranded	7.95	—
Letter Wizard	—	34.95
Mask of the Sun	—	28.75
Drol	—	25.75
Spell Wizard	—	34.95
B.C'S Quest for Tyres	9.95	—
Swag	8.95	10.95
Soldier Fortune	8.95	—
Zaga Mission	7.95	9.95
Ant Attack	8.95	—
Espionage	8.95	—
Mr Robot	8.95	11.95
Spitfire Ace	9.95	12.95
Beach Head	9.95	12.95
Dallas Quest	—	14.95
Bruce Lee	14.95	14.95
Knights of the Desert	14.95	14.95
Combat Leader	14.95	14.95
Tigers in the Snow	14.95	14.95
Caverns of Khafta	8.95	12.95
Blue Max	9.95	14.95
Multiplan	—	99.95
Paintbrush	ROM	19.95
Personal Accountant	25.60	—
Graphics Basic	—	25.55
Household Finance	22.85	—
Logo	—	56.25

100's more titles available. Send S.A.E. for free catalogue. Send cheques/POs to:

ZOOMSOFT

46 Huntsworth Mews, London NW1 6DB.
Tel: 01-723 0562

SOFTWARE UTILITIES

THE LAST ONE £57.50

SPECIAL OFFER PRICE! Normally £97.50. Program generator that frees you from the drudgery of writing tailor-made database/data management programs. Create commercial, sellable database programs with ease. TLO creates stand alone error-free coding in standard BASIC which can be compiled if required — PET-SPEED £50 (£45 if bought at same time as TLO). Input is in the form of one-line descriptions defining, in plain English, the user's requirements. Programs you create can be for any application and as simple or as sophisticated as you like. Very, very highly recommended.

OTHER SPECIAL OFFERS: Superbase £88, Simons Basic £45, Easy Spell (W) and Easy Mail £20 each, plus many more. (Catalogue 50p plus large SAE, free to customers).

The Six-Four Supplies Co.,
PO Box 19, Whitstable,
Kent CT5 1TS.
Access/Visa: (0227) 266289, 462600

SOFTWARE APPLICATIONS

HOME ACCOUNTS. Put your house in order! Probably home computing's best use! Comprehensive coverage of bank accounts. Credit cards, HP inbuilt accuracy check. Records all transactions. Projects cashflow for any period ahead. Available for C.16 CBM64 or VIC-20. £8.45 or *free* details from Discus Software, Freepost, Windmill Hill, Brixham TQ5 9BR. Telephone 08045 55532.

TO REACH BUYERS OF COMMODORE HARDWARE & SOFTWARE PHONE CLASSIFIED 01 - 437 0626

COMMODORE 64 SOFTWARE
GAMES AND BUSINESS USE

Also new releases list every 2 weeks. Send your name and address to go onto our mailing list to:

M. J. Seaward (Dept Y.C.O.)
St Olaf's Road, Hill Head,
Stratton, Nr. Bude, Cornwall.
Tel: Bude 4179 up to 10 pm.

HARDWARE

CBM 64 VIC 20

Tape Transfer, Dual Cassette Interface

It is quick and easy to transfer any cassette program with **TRIGSOFT's DCL 1** interface. DUPLICATES ALL TURBOS, M/CODE, DATA FILES ETC. IF A PROGRAM CAN BE LOADED IT WILL BE TRANSFERRED EXACTLY ON A SECOND DATASETTE.

Send £10.00 inc. P&P to: **Trigsoft, Marshfield House, 4 Burnett Place, Bradford BD5 9LX.** 7 day money back guarantee if not 100% satisfied.



TODDLERS!

Play with your parent's CBM 64. Animated musical flashcard alphabet and spelling program, developed in conjunction with school and educational psychologist. Suitable 1½-7 years.
Cassette **£5.00** Disc **£7.50**
Toddlersoft, 1 Catlane Cottage, Ewelme, Oxon OX9 6HX.
Full refund if not delighted.

FOR HIRE

VIC-20 / CBM-64 SOFTWARE HIRE

NO MEMBERSHIP FEE

Send 2 x 17p stamps for your hire kit to:
VSH (YM), 242 RANSOM ROAD, MAPPERLEY, NOTTINGHAM.
Please state which machine.



commodore 64

Alice IN VIDEOLAND



PURE MAGIC!

Join Alice in her journey through Videoland - an enchanted place populated by strange creatures such as bread-and-butterflies and pipe smoking caterpillars; where little girls change size and flamingos turn into croquet mallets!

Alice in Videoland is a revolutionary new concept in entertainment for the Commodore 64, incorporating some of the finest graphics ever seen on any home computer, accompanied by a charming musical score. There are four different game scenes involved, and your performance in earlier ones will affect your ability to get through later ones and determine your eventual total score.

Scene One - Stunning title page graphics give way to the first game scene as Alice falls into the rabbit's warren. Score points for collecting the objects to be found there - including keys to open doors, bottles to make her smaller, cakes to make her bigger!

Scene Two - Out in the garden the Cheshire cat looks on as Alice meets the pipe-smoking caterpillar. Help her to catch the bread-and-butterflies and the rocking-horse flies that change into the balls used in the croquet game in the last scene!

Scene Three - Alice is a pawn in the chess game where her opponents are the Jabberwocky and Tweedledum and Tweedledee. Help her across the board by protecting her with your White Knights!

Scene Four -The most bizarre croquet game ever! Help Alice hit the balls through the playing-card-soldier hoops before the Queen of Hearts stomps on them!

Alice in Videoland is available for the Commodore 64 on disk - £12.95, and now on cassette - £8.95.

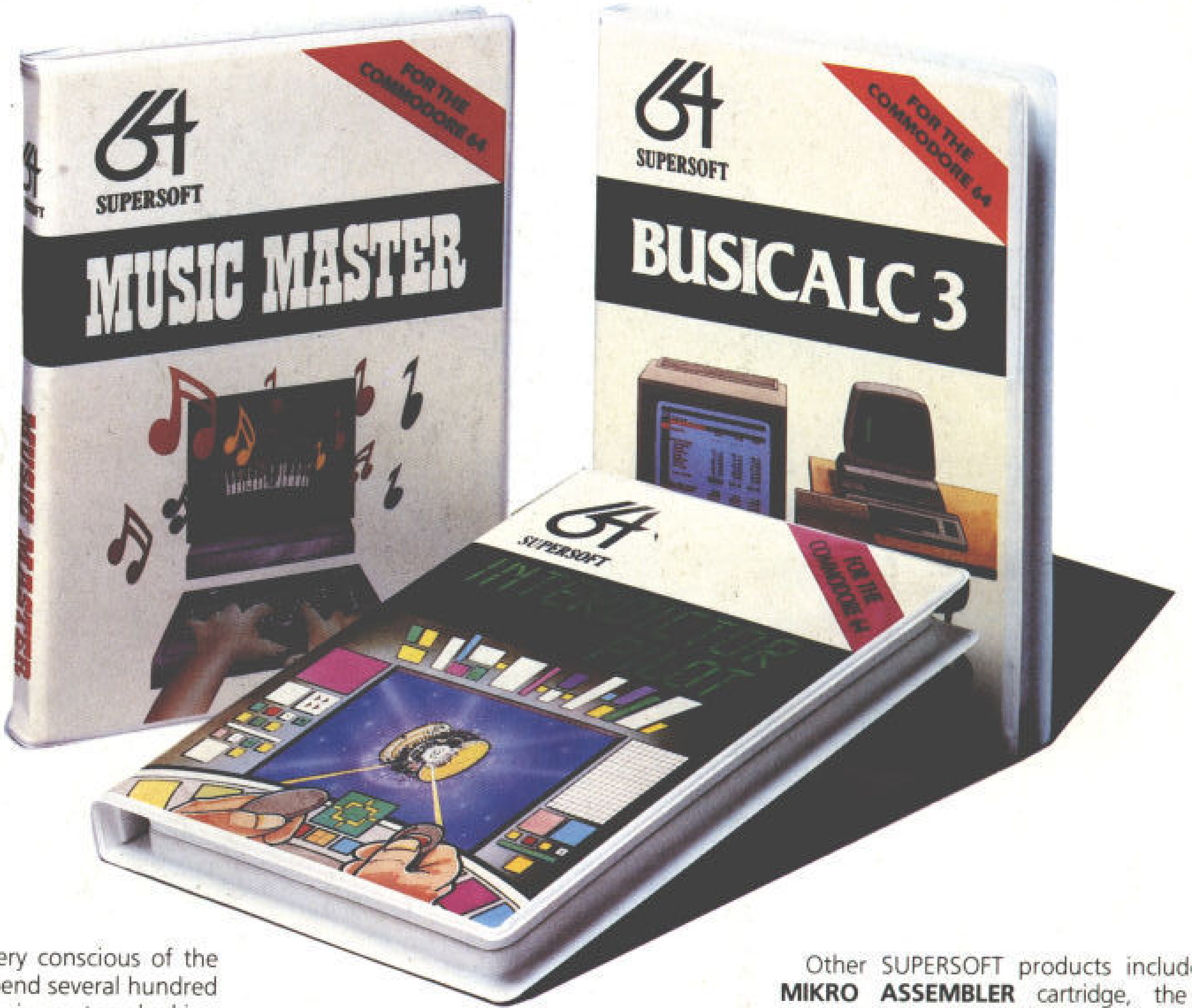
Alice in Videoland features graphics created with the Koala Pad.

Audiogenic LTD

P.O. BOX 88, READING, BERKS.

SEND FOR FREE COLOUR CATALOGUE!

Choosing the right computer is a good start — but can you find the right software?



At SUPERSOFT we're very conscious of the fact that people who spend several hundred pounds on computer equipment are looking to do rather more than play Space Invaders.

Financial planning is a rather grand name for something you've been doing all your life – making ends meet! Perhaps if Mr Micawber had used **BUSICALC** he would have been able to balance the books a little better.

For home, club or small business use BUSICALC 1 should pay for itself in no time at all; for larger companies we recommend BUSICALC 3, one of the few really valuable programs that you can learn to use in a day.

Although your Commodore 64 is a powerful musical instrument you need to be a pretty good programmer to understand how it all works. Unless, of course, you buy **MUSIC MASTER!**

To use MUSIC MASTER requires no prior musical knowledge, though in the hands of an experienced musician it will prove an invaluable tool. You don't need to know the first thing about programming either! MUSIC MASTER is the musical equivalent of a word processor, remembering the notes you play and allowing you to replay and edit them as you wish.

INTERDICTOR PILOT is a space flight simulator. Nowadays simulators are widely used to train pilots and astronauts because – to be frank – it's a lot cheaper (and safer) than the real thing!

Imagine, if you will, life in the 22nd century: space travel is commonplace, and on the outskirts of the galaxy the first war between civilizations is being fought. A shortage of trained pilots has prompted the Federation to develop a computer simulation that allows raw recruits to gain experience without paying for their mistakes with their lives. With the aid of your Commodore 64 you too can learn to pilot the Interdictor Mk 3 craft. But be warned – this is no game!

Other SUPERSOFT products include the **MIKRO ASSEMBLER** cartridge, the only assembler that's ideal for beginners yet powerful enough for the professional (most of our competitors use it!). The **VICTREE** cartridge adds dozens of commands to Basic including toolkit aids and disk commands; or on disk there's **MASTER 64,** a really comprehensive package for the keen programmer.

Of course, we do also publish games programs, and with classics like **STIX, QUINX** and **KAMI-KAZE** in our range we are one of the market leaders. But we most enjoy coming up with the sort of programs that are going to be in use for months and years, not hours and days – the sort of programs that make you glad that you bought a computer – and glad that you bought SUPERSOFT!

You won't find SUPERSOFT products on the shelves of your local supermarket. But most specialist shops stock titles from our extensive range (and are prepared to obtain other programs to order). However you can also buy direct by sending a cheque (pre-paid orders are post free!), by calling at our offices, or over the telephone using your ACCESS card.

SUPERSOFT

SUPERSOFT, Winchester House, Canning Road, Wealdstone, Harrow, Middlesex HA3 7SJ
Telephone: 01-861 1166